Tech Memo
Mat/Str 1207
ICAF 1866

Tech Memo
Mat/Str 1207
ICAF 1866

DEFENCE
RESEARCH
AGENCY

Technical Memorandum

November 1992

A Computer Program to Calculate Growth Rates for Cracks at Notches in Regions of Residual Stress

by

R. Cook
S. Dorey
D. P. Rooke

Farnborough, Hampshire

CONDITIONS OF RELEASE

0143801 316512

** DRIC U

** DRIC Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

DEFENCE RESEARCH AGENCY

Farnborough

Technical Memorandum Materials and Structures 1207

Received for printing 24 November 1992

A COMPUTER PROGRAM TO CALCULATE GROWTH RATES FOR CRACKS AT NOTCHES IN REGIONS OF RESIDUAL STRESS

by

R. Cook

S. Dorey

D. P. Rooke

SUMMARY

This Memorandum describes a computer program which can be used to predict the growth rates of fatigue cracks emanating from notches and growing through residual stress fields. The residual stress distributions, alternating loading conditions and specimen geometry must be specified by the user. The program uses a Green's function technique to calculate the stress intensity factor due to the applied and residual stresses for any crack length. The calculated stress intensity factor can be corrected to account for the exact crack shape, if it is known. The crack growth rate is obtained from a database of experimentally determined crack growth data as a function of stress intensity factor, for a number of different materials.

LIST OF CONTENTS

1 INTRODUCTION

The most common sites of fatigue damage in aircraft structures are fastener holes. The use of life enhancement techniques such as cold expansion and interference fit fasteners are now commonplace in order to combat this problem. Cold expansion increases the fatigue endurance of components by reducing the rate at which fatigue cracks grow. This is achieved by inducing a compressive residual stress field around holes in the region through which the fatigue cracks must grow. The extent of the compressive residual stress zone is typically only a few millimetres and outside this resides a balancing tensile residual stress zone. Most of the fatigue life of a component however is consumed in the growth of small cracks and therefore reducing the growth rate of such cracks can result in significant increases in overall fatigue endurance.

The design of new aircraft structures or repair schemes is now frequently performed using a damage tolerant approach. Initial defects of specific sizes are assumed to exist at all stress concentrations such as fastener holes. Fracture mechanics analyses are used to calculate the fatigue life consumed in growing a crack of the assumed initial defect size to a size which can be reliably detected by inspection. Damage tolerant design applied to cold expanded holes therefore requires that the growth rates of cracks subjected to both residual and applied stresses can be calculated.

This Memorandum describes a method of calculating the growth rates of cracks subjected to cyclic loading and growing from open holes through arbitrary residual stress fields. It also describes a computer program which can be used to perform these calculations from information input by the operator. A Green's function technique is used to calculate stress intensity factors at crack lengths specified by the user. Residual and applied stresses and specimen geometry also need to be input by the user. The total stress field is calculated by summing the residual and applied stresses and stress intensity factors are calculated by integrating the product of the stress distribution and the Green's function over the crack length. The Green's function, which is described in Ref 1, is appropriate to a through-the-thickness crack in an infinite sheet; corrections need to be made for the more commonly occurring corner crack. Various corrections for crack and specimen geometry are available in the program and are described in this Memorandum. Crack growth rates are calculated from a materials database for BS L65 and BS L71 aluminium alloys which is included in the program and can be added to by simple modifications. The method of operation of the computer program is described and the results of a test case presented.

2 THEORETICAL BACKGROUND

2.1 The prediction method

This section outlines the method used to obtain crack growth rates from a knowledge of the residual and applied stresses. Residual stress distributions induced prior to fatigue loading need to be determined and stored in the computer. Section 2.2 describes the main features of residual stress distributions, how they are specified and how the program interpolates values at any location. The applied stress distribution is calculated by the program as described in section 2.3. The total stress distribution (residual plus applied) can thus be derived and stress intensity factors calculated with the use of an appropriate Green's function. In section 2.4 a specific Green's function is described which is appropriate to a through crack growing from a central notch in an infinite component. This Green's function is used in conjunction with the residual and applied stresses to give the required stress intensity factor (K) solution. The numerical integration technique used to give the K solution is described in section 2.5. Since the Green's function is appropriate to a through crack in an infinite sheet, the K solution will in general need correcting to account for crack and component geometries. The corrections available in the program are described in section 2.6. Having obtained the required K solution it only remains to calculate the appropriate crack growth rates at specified crack lengths. The method of calculation is described in section 2.7 along with a description of the database used.

2.2 Residual stress fields and their interpolation

The residual stress fields considered in this Memorandum are appropriate to typical stress fields induced around open holes by prestressing or cold expansion. The fields considered were obtained from a number of sources such as finite element analyses, mathematical models describing the stress fields, or simple linear approximations. Whichever source was used, the distributions have certain features in common as illustrated in Fig 1a. These are:

(a) a compressive residual stress at the hole edge,

(b) the residual stress decreases in magnitude with distance away from the hole becoming tensile and reaching a maximum a short distance from the hole edge,

(c) the tensile residual stress then decreases with distance from the hole,

(d) the magnitude of the maximum compressive stress is significantly greater than the magnitude of the maximum tensile stress.

Some of the distributions (see Fig 1b) also display a region referred to as a compressive yield zone near the hole edge where the compressive stress initially increases in magnitude with distance from the hole. This occurs if the compressive stress at the hole exceeds the compressive yield point of the material.

Residual stress fields are usually described as a set of values at fixed coordinate points. It is necessary to employ an interpolation routine to calculate values between those specified and a Lagrangian technique suitable for use on a computer was selected. Unfortunately the interpolation was not sufficiently accurate at "certain" points in the distribution, for example at sharp changes of gradient or points of inflection, so that the residual stress distribution had to be split into regions. The typical residual stress distribution, shown in Fig 1a, is split into three regions which enables accurate interpolation. The residual stress distribution in which compressive yielding has occurred, shown in Fig 1b, must be split into at least four regions.

2.3 Stress intensity factors

A stress intensity factor is a single parameter which describes the magnitude of the stress field in the vicinity of a crack tip. The assumption is usually made that the behaviour of a crack is controlled by the stresses around its tip, hence stress intensity factors are very useful in characterising crack behaviour in an elastic body and are widely used for predicting the rate of growth of fatigue cracks. The stressing mode appropriate to a stress intensity factor is usually denoted by K_N, where $N = I$, II or III specifies the direction of relative movement of the crack faces as opening, sliding or tearing respectively. The cracks with which this report is concerned are opening mode cracks ($N = I$) growing along a line perpendicular to the remotely applied stress.

The stress intensity factor for the situation of a through crack of length L measured from the edge of a circular hole of radius R in an infinite sheet (see Fig 2) is given by:

$$K_I = 1/\sqrt{\pi L} \int_0^L \sigma(x) G(L/R, x/L).dx \quad , \tag{1}$$

where $\sigma(x)$ is the stress distribution along the crack site in the uncracked configuration,

$G(L/R, x/L)$ is the appropriate Green's function for this configuration.

The stress distribution along a line from a hole in an infinite sheet perpendicular to a remotely applied stress is given by:

$$\frac{\sigma_A}{S} = \frac{1}{2}\left(2 + \left[\frac{1}{1 + x/R}\right]^2 + 3\left[\frac{1}{1 + x/R}\right]^4\right) , \tag{2}$$

where R = radius of the hole

x = distance along the crack

S = remotely applied stress

The stress intensity factor distribution for this stress field is known. Values of stress intensity factor were calculated with no residual stress field present and compared with the known solution to test the accuracy of the computer program. If a residual stress field is present in the component then the stress distribution is given by the sum of the applied and residual stress distributions.

2.4 The Green's function

The Green's function used was developed by Shivakumar and Forman[1] and is for a through crack at the edge of a circular hole in an infinite sheet.

The Green's function $G(L/R, x/L)$ can be expressed as $G(\lambda,\beta)$, that is,

$$G(\lambda,\beta) = \left(\frac{2\beta}{1-\beta}\right)^{1/2} + C(\alpha,\beta)\left[2\frac{(1+F_\beta)}{(1-\beta^2)^{1/2}} - \left(\frac{2\beta}{1-\beta}\right)^{1/2}\right], \tag{3}$$

where $\lambda = L/R$, $\beta = x/L$ and $\alpha = 1/(1+\lambda)$, and

$$F_\beta = (1-\beta^2)(.2945 - .3912\beta^2 + .7635\beta^4 - .9942\beta^6 + .5094\beta^8) \tag{4}$$

and

$$C(\alpha,\beta) = \sum_{m=1}^{5}\left(\sum_{n=0}^{3} C_{m,n}\alpha^{m/2}\beta^{n/2}\right), \tag{5}$$

where $C_{m,n}$ are constants whose values are given in Table 1.

Table 1

Coefficients in Green's function

m/n	0	1	2	3
1	0.8164	−4.5911	7.6059	−3.8529
2	0.0492	17.3181	−36.8465	20.6753
3	−0.4831	−30.5563	75.4833	−44.3540
4	−0.1746	26.2877	−73.1167	44.2607
5	0.7952	−8.4570	26.8666	−16.7296

2.5 Numerical integration

It can be seen that the Green's function is algebraically complex and for this reason, as well as the fact that the residual stress distribution is unlikely to be represented by a simple polynomial, the integration has to be carried out by numerical means. The method selected is a Gaussian quadrature numerical integration method, chosen because not only is it very accurate but also because, using suitable weighting functions, it is able to take account of the singularity which occurs, due to the $\sqrt{1-\beta}$ and $\sqrt{1-\beta^2}$ denominators, in the Green's function as $\beta \to 1$.

Gaussian integration is carried out as follows:- Given a weighting function $W(Y)$ and a function $F(Y)$, there exists a Gaussian formula of the form

$$\int_a^b W(Y)\,F(Y)\,dY \simeq \sum_{i=1}^{n} h_i . F(Y_i) \ , \qquad (6)$$

where Y_i are the zeros of the appropriate nth degree orthogonal polynomial, and h_i are the corresponding quadrature coefficients. Both Y_i and h_i are tabulated for various weighting functions[2].

For the purposes of this work, the function $F(Y)$ is the product of the reduced Green's function $g(\lambda, \beta)$ and the total stress distribution, that is $\sigma(x)g(\lambda, \beta)$, and the weighting function is $1/\sqrt{(1-\beta)}$, where $g(\lambda, \beta)$ is defined as

$$g(\lambda, \beta) = \sqrt{1-B}\; G(\lambda, \beta) \qquad (7)$$

An eighth order integration ($n = 8$) was chosen, as it gave an acceptable compromise between running time and accuracy. Obviously, the greater the number of summations that are carried out, the longer will be the processing time but also the greater the accuracy of the process. However, it was found that no significant increase in accuracy was attained by carrying out a higher order integration than eight, so the increase in processing time required to carry out such a process was thought to be unnecessary.

Because of the shape of some of the residual stress fields under consideration, it was necessary to carry out the integration in regions, as described in section 2.1. This is due to the fact that the interpolation method was unable to take sufficient account of sharp changes in gradient of the stress distribution; the distribution was therefore interpolated in sections.

To this end the following scheme was worked out:-

The integral expression for K_1 in equation (1) can be written

$$K_I = \sqrt{\frac{L}{\pi}} \int_0^1 \frac{\sigma(L,\beta)g(\lambda,\beta)d\beta}{\sqrt{1-\beta}} \ . \qquad (8)$$

The distribution is split into n regions and the boundaries between regions are denoted by $L_1, L_2, \ldots L_n$ with $\beta_j = L_j/L$ for $j = 1, 2 \ldots (n-1)$; $\beta_0 = 0$ and $\beta_n = 1$. The width of a region is given by $\Delta_j = \beta_j - \beta_{(j-1)}$. Thus, in general,

$$K_I = \sqrt{\frac{L}{\pi}} \sum_{j=1}^{n} I_j \,, \tag{9}$$

where

$$I_j = \int_{\beta_{j-1}}^{\beta_j} \frac{\sigma(L,\beta)g(\lambda,\beta)}{\sqrt{1-\beta}} \, d\beta \,. \tag{10}$$

Substitution of $y = (\beta - \beta_{j-1})/\Delta_j$ leads to

$$I_j = \sqrt{\Delta_j} \int_0^1 \frac{\sigma(L, y\Delta_j + \beta_{j-1})g(\lambda, y\Delta_j + \beta_{j-1})}{\sqrt{\dfrac{1 - y\Delta_j - \beta_{j-1}}{\Delta_j}}} \, dy \,, \quad j = 1,2,\ldots(n-1)$$

and (11)

$$I_n = \sqrt{\Delta_n} \int_0^1 \frac{\sigma[L(y\Delta_n + \beta_{n-1})]g(y\Delta_n + \beta_{n-1})}{\sqrt{1-y}} \, dy \,.$$

The numerical integrations for $I_j (j = 1,\ldots n-1)$ were done using Gaussian quadrature with unit weighting function as the integrand is not singular. Integration of I_n was done using Gaussian quadrature with a weighting function of $(1 - y)^{-1/2}$ to allow for the singularity at $y = 1$.

2.6 Corner crack corrections

As has already been observed, the stress intensity factors obtained by the Green's function approach, are for through cracks in an infinite sheet. In order to obtain stress intensity factors for the more commonly occuring corner cracks in a sheet of finite dimensions, correction factors need to be applied. The correction factors relate to the shape of the crack and the width and thickness of the sample in which the crack is growing. Both correction factors (crack shape and finite width) are based on the work of Newman and Raju[3] who developed empirical equations for stress intensity factors based on finite element analyses. The equations developed are relevant to quarter-elliptical corner cracks in finite width components. They account for both component width and thickness, and the shape of the crack in terms of its aspect ratio (bore length/surface length). Since the Newman and Raju work is for the case of no residual stress, then the correction factors required cannot be directly determined. Two methods are used to determine the corner crack shape corrections and one method to determine the finite width correction.

Firstly examining the factors in equations (50-63) of Ref 3, it is apparent that the g_2 factor in Ref 3 accounts for the presence of the hole in calculating the stress intensity factor. It is very similar in magnitude to the factor derived using the Shivakumar and Forman method assuming no residual stress. The first correction factor offered in the present program is the solution calculated by Newman and Raju with the g_2 term set to 1. This correction factor is multiplied by the Shivakumar and Forman solution for a through crack in a residual stress field. This gives the solution for an elliptical crack in an arbitrary residual stress field in an infinite width component. This will subsequently be referred to as the G2 correction. The remaining terms in the equations refer to the geometries of the test piece and crack. The crack aspect ratio and the specimen thickness have to be input by the user. This method assumes that the g_2 term entirely accounts for the presence of the hole and that there is no 'fine tuning' included in the other terms in order to achieve a good fit to the finite element solutions.

The second corner crack correction factor is obtained using the complete Newman and Raju solution, *ie* including the g_2 term. The solution again requires the input of both crack aspect ratio and component thickness, by the user. The complete Newman and Raju solution is divided by the Shivakumar and Forman solution assuming no residual stress field is present. This results in a correction factor which when multiplied by the Shivakumar and Forman solution with a residual stress field present, gives a solution for an elliptical crack in a residual stress field in a component of infinite width.

There is also an option in the program to correct for the finite width of the component. The correction factor available is that due to Newman and Raju and is given by equation (47) of Ref 3.

2.7 Crack rate predictions

Crack rates are calculated by comparing the derived stress intensity factors with known crack rate data. The program calculates the maximum and minimum stress intensity factor values and hence the values of ΔK and stress intensity factor ratio R. A subroutine interpolates or extrapolates as appropriate from a database of crack rates versus stress intensity factors at a range of R values for a specific material to calculate crack growth rate. The routine used in the program is that developed by Edwards[4]. The database used is also the one described in Ref 4 and is based on data obtained using BS L71 and BS L65 aluminium alloys. The data is contained in Lines 10150 to 10290 in the program. Other data may be substituted in the correct form (see Ref 4), or the program modified to access an alternative database of crack growth rates.

3 THE PROGRAM

3.1 Program outline

The program was written in Hewlett Packard BASIC (BASIC 2.0) and is listed in the Appendix. An outline of the scope of the program is described in this section. Details of how to run the program are given in section 3.2 which can be used as a users guide. The file format required for the residual stress data is described in section 3.3. This format must be used if the user creates data files which can be accessed by the program.

The program will calculate stress intensity factors at crack lengths specified by the operator, taking into account a residual stress field if required. When running the program, the operator will need to input the following information:

(a) Residual stress field: the residual stress field can be defined either by inputing coordinates via the keyboard or by loading data from a pre-stored file, the format of these files is given in section 3.3. It is also possible to run the program with no residual stress field.

(b) Crack lengths: stress intensity factors can be calculated either over a range of crack lengths at regular increments or for a number of crack lengths specified by the operator.

(c) Dimensions of the component and crack aspect ratio: stress intensity factors can be corrected for the component dimensions (thickness and width) and the crack aspect ratio.

(d) Fracture toughness K_{Ic}: the fracture toughness is required to determine when failure of the test piece will occur.

Stress intensity factor values will be calculated at the specified crack lengths and printed out. The program will then calculate the crack growth rates corresponding to these stress intensity factors.

The operator is given several final options such as re-running the program with the same residual stress field, storing the residual stress field, stress intensity factors and rate values for future reference, plotting current data and plotting previously stored distributions against one another for comparison.

3.2 Running the program

A description is given below of how to run the program, including all of the options. The prompts given by the program are written in block capitals and responses in parentheses. The running procedure is split into six sections (a)-(f). These sections are shown in the flow diagram in Fig 3.

(a) Inputing residual stress fields (RSF)

(RUN)

MULTI-PLOT (Y/N) – the operator can elect to run the routine to plot previously stored distributions. The Y response is described at the end of this section. The N response gives the following prompt.

INITIAL OPTIONS (Opt1)

0. NO RESIDUAL STRESS FIELD

1. INTERPOLATED RESIDUAL STRESS FIELD

2. RESIDUAL STRESS FIELD LOADED FROM FILE (0,1 or 2)

0. to run program without any residual stress field, *ie* with applied stress only.

1. to input a residual stress field from the keyboard.

2. to input a residual stress field from a previously stored file.

Whichever option is taken, the program next asks for the applied stress range:

MAX REMOTE STRESS (MPa)

MIN REMOTE STRESS (MPa)

If Opt1 was 0, then the prompt;

RADIUS OF THE HOLE (mm) is given and the operator should input the radius of the hole. As no RSF is to be input, the program goes on to the next section (*).

If Opt1 was 1, then the prompts to input the residual stress field (RSF) are as follows:

RADIUS? (mm) – operator inputs the radius of the hole,

ORIGIN OF DISPLACEMENT COORDS. (0 OR 1)

0 – FOR CENTRE OF HOLE.

1 – FOR EDGE	– operator inputs the origin used when describing the distances in the residual stress distribution.
VALUE OF X? (mm)	– operator inputs a pair of coordinates of the RSF: the position and magnitude of the residual stress.
VALUE OF STRESS? (MPa)	– the program will continue to give these prompts after each pair of coordinates have been input until a single carriage return is input for X.

The coordinates are printed on the screen as they are input and each pair given a number. When the input loop is terminated, the prompt;

CHANGE ANY? (Y/N) is given. The operator has the chance to change any pair of coordinates which may have been erroneously entered. If there are entries to be changed (Y), the prompt;

ENTRY TO BE CHANGED? is given. The operator should enter the number of the coordinate pair which are to be altered. The operator will then be asked to repeat the entry number and should enter the revised coordinates. The operator will again be asked if their are any inputs to change and this will be repeated until a negative response (N) is given.

STORE DISTRIBUTION? (Y/N). The operator can choose to store the RSF in a tile for future use. If so;

CREATE NEW FILE? (Y/N). The RSF can be stored either in a new file or in a file that already exists.

FILE NAME? The operator should input the name of the file to be used/created.

If Opt1 had been 2 then the operator would be asked;

NAME OF FILE? The operator inputs the name of the file containing the RSF to be used.

At this point, the Opt1 = 1 and Opt1 = 2 sequences become coincident.

(b) Plotting and checking of residual stress fields

The coordinates of the distribution are printed on the screen. Whenever a plot is to be made, from any part of the program, the operator will be given the range of coordinates on both axes and asked to define the size of the intervals within these ranges at which to print the grid dimensions.

X-AXIS LABEL SPACING? – Operator inputs the interval at which grid dimensions on X and Y axes are to be printed.

Y-AXIS LABEL SPACING? – The distribution is plotted, after which the operator is given the chance to obtain a hard copy.

NO. OF REGIONS – The operator must decide on the number of regions into which the distribution is to be split.

INPUT NUMBERS OF POSITIONS OF BOUNDARIES.

POSITION OF END OF REGION N (where N = No. of regions; $N \geq 1$)

The coordinates of the RSF values are printed out and numbered so that the operator can use the numbers to indicate where the boundaries between the regions are to be inserted.

The distribution is now re-plotted this time using data interpolated by the program. The operator is thus able to see if the interpolated data accurately represents the original input data. The operator is asked;

OK? (Y/N) and should indicate whether or not the data are satisfactory.

If not, the program allows the operator to re-position the boundaries in order to produce a better result. The program therefore returns to the NO. OF REGIONS prompt.

(c) Input of crack lengths

(*) Now that the RSF has been input (or the no RSF option taken) the program goes on to request the crack lengths at which stress intensity factors are to be calculated.

1 SIFs CALCULATED OVER A RANGE OF CRACK LENGTHS

2 SIF CALCULATED FOR SPECIFIC CRACK LENGTHS

If option 1 (OPT2=1) is taken, the operator will be asked to input the range over which the SIFs are to be calculated, thus;

START OF RANGE. (mm)

END OF RANGE. (mm)

INCREMENTAL LENGTH. (mm)

If option 2. is taken then the prompt;

CRACK LENGTH? is given and the operator can input one or a list of crack lengths. The operator must input a crack length followed by a carriage return. When all crack lengths have been input, the response to the CRACK LENGTH prompt should be carriage return.

(d) Correction factors

The operator is now asked to decide on the type of correction routine to be used. The options are;

0. NO CORRECTION

1. NEWMAN & RAJU WITHOUT G2

2. N&R/SHRIV AS CORR

For an explanation of the two corrections, see section 2.6.

Unless option 0. is taken, the dimensions of the test piece will be required;

THICKNESS? (mm) – Operator inputs specimen thickness

A/C? – Operator inputs aspect ratio of the crack, where A = length down the bore,
C = length along the surface.

DO YOU WANT A WIDTH CORRECTION? (Y/N) – operator is given the option of leaving out the width correction.

WIDTH? (mm) – if required.

(e) Stress intensity factor and crack growth rate calculations

The program now calculates and prints out the maximum and minimum stress intensity factors and also $\Delta K (= K_{max} - K_{min})$ and $R (= K_{min}/K_{max})$. After each printout, the option is given of obtaining a hard copy. The operator has the option of plotting K_{max} vs. L and/or ΔK and R vs. L via the prompts.

PLOT KMAX? (Y/N) and

PLOT KMAX – KMIN AND R? (Y/N)

The operator is then asked whether rate values are to be calculated;

CALCULATE RATE VALUES? (Y/N). If the operator responds in the affirmative, he is given the prompt;

KC VALUE? ($MNm^{-3/2}$) and should input the K_{Ic} value for the material under consideration. The program then calculates and prints out the predicted rate values corresponding to the calculated stress intensity factors. The option is given for a hard copy of the rate values and for a plot of rate vs. crack length.

The operator is now asked whether or not the distributions (RSF, SIF's and rates) are to be stored for future reference and/or comparison with other distributions. If the response is yes, then the operator can either store them in an existing file or create a new one, giving the name of the file to be used/created.

STORE DISTRIBUTIONS? (Y/N)

CREATE NEW FILE? (Y/N)

FILE NAME?

(f) End options

ANOTHER RUN SAME DIST? (Y/N). This allows the program to be re-run with the same residual stress distribution, for example if different crack lengths are to be used. If the response is yes (Y), the program returns to (*).

If not (N), the operator is now given the prompt;

PLOT GROUPS OF DIST'S? (Y/N). This enables the operator to plot one or more distributions of residual stress fields, stress intensity factors or crack growth rates from previously stored files.

NAME OF FILE? The operator should enter the name of one of the files to be used.

ANOTHER FILE? (Y/N). If another file is to be read the operator enters Y and the program asks for a file name again. This process is repeated until no more files are to be read and the operator answers N.

These options are then given;

0 – EXIT

1 – RESIDUAL STRESS FIELD

2 – STRESS INTENSITY FACTOR DIST.

3 – CRACK GROWTH RATE DIST.

The operator can choose which of the three distributions to plot and they will then be plotted on the same axes for all the files selected,

Option 0. ends the program.

3.3 File format

Residual stress fields may be input via the keyboard, as described in section 3.2, or may be accessed by the program from a data file. It is important when creating a data file to ensure that it has the correct format. Details of the file format required are given below.

The general form of the data file is:-

Lz (1,0)
Lz (2,0)
Lz (3,0)
Lz (0,1), Lz (0,2)
Lz (1,1), Lz (1,2)
Lz (2,1), Lz (2,2)
Lz (n,1), Lz (n,2)

Where Lz (1,0) describes the coordinate system and is equal to
0 if the X coordinates are measured from the centre of the hole
1 if the X coordinates are measured from the edge of the hole
Lz (2,0) = No of entries in the file array
Lz (3,0) = radius of the hole
Lz (0,1) to Lz (n,1) are the distances from the defined origin
Lz (0,2) to Lz (n,2) are the residual stresses at the corresponding positions

The maximum dimension of the array is (50,2), if more than 50 coordinates are used, then the dimension statement (line 7920) must be altered.

4 EXAMPLE - A PRESTRESSED HOLE

An example of the use of this program is presented in this section to illustrate its capabilities. The example chosen is that of a BS 2L65 aluminium alloy test specimen containing a central hole (see Fig 4). Residual stresses may be induced in the component by applying an axial load large enough to cause local yielding around the central hole. A number of specimens were subjected to such prestress treatments and then to cyclic fatigue loading of 110 ± 96.5 MPa, when crack growth measurements were made[5]. The crack growth data measured in the experimental investigation will be compared with the crack growth data predicted using this computer program. The specimen geometry and loading conditions which need to be input to the program are therefore defined as those used in the experimental investigation. The residual stress fields resulting from various prestress levels were determined using a finite element analysis. Other inputs required by the program will be described later in this section.

Three values of prestress were used in the experimental investigation resulting in three different residual stress distributions across the minimum section of the components. The values of prestress used were $a = 60\%$ of the yield stress on the net section, $b = 70\%$ and $c = 80\%$, resulting in local yielding in all cases at the edge of the hole. Residual stresses determined using a finite element program are plotted in Figs 5, 6 and 7 for the three prestresses ((a), (b) and (c)) used in the experimental investigation. Each distribution was split into three or more regions and the polynomial expressions derived by the program are also shown in these figures as solid curves.

Stress intensity factors were evaluated at crack lengths covering two ranges; 0.1 to 0.5 mm in steps of 0.1 mm and 1 to 6 mm in steps of 0.5 mm. The ranges covered crack lengths at which experimental crack growth data had been measured. Corrections to the stress intensity factor were made using both of the corner crack routines in conjunction with the finite width correction (described in section 2.6). The aspect ratio selected was $A/C = 1.4$ which was thought to most accurately represent what occurred in practice. Crack growth rates were then calculated for each of the prestress levels, crack length ranges and correction factor routines described above. The predicted rates are shown for the three prestress levels in Figs 8, 9 and 10. The experimental data are also shown as solid bars representing the range of measured crack growth rates. The dots on each of the solid bars represent the log mean value of the experimental data.

5 DISCUSSION

The residual stress distributions input to the program consisted of eleven coordinate values and are shown in Figs 5 to 7. The polynomials fitted in segments by the computer program were a reasonable interpretation of the input data. However due to the small number of coordinates specified, abrupt changes in slope of the fitted curves occur near to the region boundaries where the data is sparse. It is expected that in most applications a larger number of residual stress coordinates will be input to the program enabling better data fits. Predicted crack growth rates using the two different correction routines (Figs 8 to 10) are in reasonable agreement with each other and with the experimentally measured data. This gives some confidence in the methods used but does not give an indication of which correction method should be recommended.

There are two main criticisms of the approach used in this paper. Firstly it is assumed that the residual stress field present in an uncracked body remains unaltered by the presence and growth of fatigue cracks. This is clearly not possible and the residual stress field must redistribute as cracking occurs. This aspect is currently being investigated. Some modification to the models used in the program may be possible, dependent on the complexity of the redistribution process. The second main criticism is that closure of the crack flanks is not taken into account. Calculation of stress intensity factors is based on the range of the total local stresses and takes no account of the fact that the crack may be closed due to compressive residual stresses acting behind the crack tip. The stress intensity factor range experienced by the crack tip will be different from that predicted using the methods described.

Two further criticisms of the method are also being considered. The first is that the majority of the fatigue life of a component containing beneficial compressive residual stresses in the area of crack initiation, will be dominated by short crack growth. It has been shown[6] that short cracks propagate faster than long cracks when subjected to the same stress intensity factor range calculated by conventional linear elastic fracture mechanics, the effect being most marked at negative stress ratios. The crack rate database contained in the program was derived from long crack data but is used to predict the growth rates of short cracks. From the foregoing it is clear that predicted growth rates at short crack lengths may be underestimated particularly since the presence of compressive residual stresses leads to highly negative stress ratios. Short crack growth rate databases are being generated for future use in the program. The second criticism is that the Green's function solution used in the program is not applicable to most engineering situations as cracks are not generally through the thickness and the components are obviously not infinitely wide. The corrections needed to modify the calculated stress intensity factors could lead to considerable errors in the predictions if they are not entirely appropriate. Other Green's function solutions are currently being sought and determined.

6 CONCLUSIONS

(1) A method has been devised, and a computer program has been developed, for predicting the growth rates of fatigue cracks emanating from open holes and propagating though arbitrary residual stress fields.

(2) The program has been used to predict the growth rates of cracks from holes subjected to prestressing. Reasonable agreement was found between predicted and experimental results.

(3) Shortcomings of the prediction method have been identified and suitable modifications proposed.

Appendix

PROGRAM LISTING

```
100   COM /Opt/ Opt1,Opt2
110   COM /Aps/ Asmax,Asmin
120   COM /Fle/ File(2,50,2)
130   COM /Title/ Th,B,Aoc,Name$[30]
140   DIM Gauss(50,7),Rpt(50,2)
150   Th=0
160   B=0
170   Aoc=0
180   K2=0
190   Name$=""
200   PRINTER IS 1
210   OUTPUT 2;"K";
220   GCLEAR
230   CALL Arrays
240   PRINT
250   PRINT
251   Q$=""
260   INPUT "MULTI-PLOT?",Q$
320   IF Q$="Y" THEN
330   CALL Mplot
340   GOTO 3020
350   END IF
360   IF Q$="N" THEN GOTO 390
370   GOTO 251
390   PRINT
400   '
410   'INITIAL OPTIONS
420   PRINT "OPTIONS;"
430   PRINT
440   PRINT "0.  NO RESIDUAL STRESS FIELD."
450   PRINT
460   PRINT "1.  INTERPOLATED RESIDUAL STRESS FIELD"
470   PRINT
480   PRINT "2.  RESIDUAL STRESS FIELD LOADED FROM FILE"
490   PRINT
491   Q$=""
500   INPUT Q$
560   ON ERROR GOTO 491
570   Opt1=VAL(Q$)
580   OFF ERROR
590   IF Opt1<>0 AND Opt1<>1 AND Opt1<>2 THEN
600   GOTO 491
610   END IF
620   INPUT "MAX. REMOTE STRESS? (MPa)",Max$
630   IF LEN(Max$)=0 THEN GOTO 620
640   ON ERROR GOTO 620
650   Asmax=VAL(Max$)
660   OFF ERROR
670   INPUT "MIN. REMOTE STRESS? (MPa)",Min$
680   IF LEN(Min$)=0 THEN GOTO 670
690   ON ERROR GOTO 670
700   Asmin=VAL(Min$)
710   OFF ERROR
720   IF Opt1=0 THEN GOTO 750
730   CALL Inputs(Radius)
731   GCLEAR
740   GOTO 810
750   Name$="NO RSF"
760   INPUT "RADIUS OF THE HOLE? (mm)",Rad$
770   IF LEN(Rad$)=0 THEN GOTO 760
780   ON ERROR GOTO 760
790   Radius=VAL(Rad$)
800   OFF ERROR
810   GCLEAR
```

```
820   OUTPUT 2;"K";
830   !
840   !OPTIONS ON THE TYPE OF RUN
850   PRINT "OPTIONS:"
860   PRINT
870   PRINT "1.    SIF'S CALCULATED OVER A RANGE OF CRACK LENGTHS."
880   PRINT
890   PRINT "2.    SIF CALCULATED FOR SPECIFIC CRACK LENGTHS."
891   Q$=""
900   INPUT Q$
950   Opt2=VAL(Q$)
960   OFF ERROR
970   IF Opt2<>1 AND Opt2<>2 THEN
980   GOTO 891
990   END IF
1010  !
1020  !
1030  CALL Sif1(Radius,Gauss(*),Lp)
1040  !
1050  Th$=VAL$(Th)
1060  B$=VAL$(B)
1070  Aoc$=VAL$(Aoc)
1080  !
1090  IF Th=0 THEN Th$="NONE"
1100  IF B=0 THEN B$="NONE"
1110  IF Aoc=0 THEN Aoc$="NONE"
1120  !
1130  !
1131  Q$=""
1140  INPUT "DO YOU WANT A COPY?",Q$
1200  IF Q$="Y" THEN GOTO 1240
1210  IF Q$="N" THEN GOTO 1490
1220  GOTO 1131
1230  !
1240  PRINTER IS 701
1250  !
1260  !
1270  PRINT Name$
1280  PRINT "STRESS INTENSITY FACTOR DISTRIBUTION."
1290  PRINT DATE$(TIMEDATE)
1300  PRINT "RADIUS= ";Radius;"mm."
1310  PRINT "APPLIED STRESS RANGE= ";Asmax;"-";Asmin;"MPa."
1320  PRINT "SPECIMEN THICKNESS= ";Th$;" mm."
1330  PRINT "SPECIMEN WIDTH= ";B$;" mm."
1340  PRINT "A/C RATIO= ";Aoc$
1350  PRINT
1360  PRINT
1370  !
1380  PRINT "CRACK LENGTH","KMAX.(MNm^-3/2)","KMIN","KMAX-KMIN","KMIN/KMAX"
1390  PRINT "          (mm)"
1400  PRINT
1410  FOR Pr=0 TO Lp
1420  PRINT USING "3X,DD.DD,14X,SD.2DE,7X,SD.2DE,4X,SD.2DE,2X,SD.DDD";Gauss(Pr,1
),Gauss(Pr,2),Gauss(Pr,3),Gauss(Pr,4),Gauss(Pr,5)
1430  NEXT Pr
1440  FOR Iou=1 TO 4
1450  PRINT
1460  NEXT Iou
1470  PRINTER IS 1
1480  !
1490  !
1491  Q$=""
1500  INPUT "PLOT KMAX?",Q$
1560  IF Q$="Y" THEN GOTO 1600
1570  IF Q$="N" THEN GOTO 1671
1580  GOTO 1491
```

```
1590  '
1600  Ult=Lp
1610  DIM Str$[30]
1620  Str$="STRESS INTENSITY FACTOR DIST."
1630  Xlabel$="L"
1640  Ylabel$="K.MAX"
1650  CALL Plt(Ult,Gauss(*),Str$,Xlabel$,Ylabel$)
1660  !
1670  !
1671  Ult=Lp
1673  OUTPUT 2;"K";
1674  Q$=""
1675  INPUT "PLOT KMAX-KMIN AND R?",Q$
1676  IF Q$="Y" THEN
1677  CALL Dplot(Gauss(*),Ult)
1678  GOTO 1684
1679  END IF
1681  IF Q$="N" THEN GOTO 1684
1682  GOTO 1674
1683  !
1684  Q$=""
1690  INPUT "CALCULATE RATE VALUES?",Q$
1750  IF Q$="Y" THEN GOTO 1780
1760  IF Q$="N" THEN GOTO 2410
1770  GOTO 1684
1780  INPUT "KC VALUE? (MNm^3/2)",K$
1790  IF LEN(K$)=0 THEN GOTO 1780
1800  ON ERROR GOTO 1780
1810  K2=VAL(K$)*(1000^.5)
1820  OFF ERROR
1830  GCLEAR
1840  OUTPUT 2;"K";
1850  PRINT "CRACK LENGTH","RATE.(mm/CYCLE)"
1860  PRINT "        (mm)"
1870  PRINT
1880  FOR Rt=0 TO Lp
1890  Srange=(Gauss(Rt,2)+Gauss(Rt,3))/(SQR(1.E-3)*2)
1900  Sdiff=(Gauss(Rt,2)-Gauss(Rt,3))/(SQR(1.E-3)*2)
1910  CALL Rate(Srange,Sdiff,Gauss(Rt,6),K2)
1920  PRINT Gauss(Rt,1),"",Gauss(Rt,6)
1930  Rpt(Rt,1)=Gauss(Rt,1)
1940  Rpt(Rt,2)=Gauss(Rt,6)
1950  File(2,Rt,1)=Gauss(Rt,1)
1960  File(2,Rt,2)=Gauss(Rt,6)
1970  NEXT Rt
1980  File(2,0,0)=Lp
1990  !
1991  Q$=""
2000  INPUT "DO YOU WANT A COPY",Q$
2060  IF Q$="Y" THEN GOTO 2100
2070  IF Q$="N" THEN GOTO 2290
2080  GOTO 1991
2090  !
2100  PRINTER IS 701
2110  !
2120  PRINT Name$
2130  PRINT "CRACK GROWTH RATE PREDICTIONS."
2140  PRINT DATE$(TIMEDATE)
2150  PRINT "RADIUS= ";Radius;"mm."
2160  PRINT "APPLIED STRESS RANGE= ";Asmax;"-";Asmin;"MPa."
2170  PRINT "SPECIMEN THICKNESS= ";Th$;" mm."
2180  PRINT "SPECIMEN WIDTH= ";B$;" mm."
2190  PRINT "A/C RATIO= ";Aoc$
2200  PRINT "K1c= ";K2;"Nmm^-3/2"
2201  PRINT
2202  PRINT
```

```
2210 !
2220 PRINT "CRACK LENGTH","RATE (mm/CYCLE)"
2230 PRINT "          (mm)"
2240 PRINT
2250 FOR Pri=0 TO Lp
2260 PRINT Gauss(Pri,1),"",Gauss(Pri,6)
2270 NEXT Pri
2280 PRINTER IS 1
2290 !
2291 Q$=""
2300 INPUT "PLOT?",Q$
2360 IF Q$="Y" THEN GOTO 2390
2370 IF Q$="N" THEN GOTO 2410
2380 GOTO 2291
2390 CALL Plt(Lp,Rpt(*),"CRACK GROWTH RATE","L","RATE")
2400 !
2410 !
2411 Q$=""
2420 INPUT "STORE DISTRIBUTIONS?",Q$
2480 IF Q$="Y" THEN GOTO 2511
2490 IF Q$="N" THEN GOTO 2791
2500 GOTO 2411
2510 !
2511 Q$=""
2520 INPUT "CREATE NEW FILE?",Q$
2580 IF Q$="Y" THEN GOTO 2640
2590 IF Q$="N" THEN GOTO 2730
2600 GOTO 2511
2610 !
2620 GOTO 2640
2630 PRINT TABXY(4,10):"FILENAME ALREADY USED."
2640 INPUT "FILE NAME?",Name$
2650 IF LEN(Name$)=0 THEN GOTO 2640
2660 ON ERROR GOTO 2630
2670 CREATE BDAT Name$,5
2680 OFF ERROR
2690 GOTO 2750
2700 OUTPUT 2;"K";
2710 PRINT TABXY(4,10):"FILE NOT FOUND. TRY AGAIN."
2720 PRINT
2730 INPUT "FILE NAME?",Name$
2740 ON ERROR GOTO 2700
2750 ASSIGN @Pth TO Name$
2760 OFF ERROR
2770 OUTPUT @Pth;File(*)
2780 ASSIGN @Pth TO *
2790 !
2791 Q$=""
2800 INPUT "ANOTHER RUN, SAME DIST.?",Q$
2860 IF Q$="Y" THEN GOTO 810
2870 IF Q$="N" THEN GOTO 2891
2880 GOTO 2791
2890 !
2891 Q$=""
2900 INPUT "PLOT GROUP OF DIST'S?",Q$
2960 IF Q$="Y" THEN GOTO 3000
2970 IF Q$="N" THEN GOTO 3020
2980 GOTO 2891
2990 !
3000 CALL Mplot
3010 !
3020 OUTPUT 2;"K";
3030 GCLEAR
3040 !
3050 END
3060 !
```

```
3070  !*************************************************************
3080  !*************************************************************
3090  !
3100  SUB Arrays
3110  !
3120  !THIS SUBROUTINE SETS UP ARRAYS WHICH HOLD ZEROS FOR GAUSS. INTEG.
3130  !AND COEFFS FOR C(ALPHA,BETA)
3140  COM /Const/ C(5,4)
3150  COM /Zer/ Ero1(8,2),Ero2(8,2)
3160  RESTORE 3250
3170  FOR We1=1 TO 8
3180  READ We
3190  Ero1(We1,1)=2*We
3200  NEXT We1
3210  FOR Ro=1 TO 8
3220  READ Er
3230  Ero1(Ro,2)=√(1-Er^2)
3240  NEXT Ro
3250  DATA 0.189451,0.182603,0.169157,0.149596,0.124629,0.095159,0.062254,0.0271
52
3260  DATA 0.095013,0.281604,0.458017,0.617876,0.755404,0.865631,0.944575,0.9894
01
3270  RESTORE 3360
3280  FOR We1=1 TO 8
3290  READ We
3300  Ero2(We1,1)=We
3310  NEXT We1
3320  FOR Ro=1 TO 8
3330  READ Er
3340  Ero2(Ro,2)=Er*.5+.5
3350  NEXT Ro
3360  DATA 0.10122,0.22238,0.31370,0.36268,0.36268,0.31370,0.22238,0.10122
3370  DATA -0.96028,-0.79666,-0.52553,-0.18343,0.18343,0.52553,0.79666,0.96028
3380   RESTORE 3440
3390   FOR Bb=0 TO 3
3400   FOR Aa=1 TO 5
3410   READ C(Aa,Bb)
3420   NEXT Aa
3430   NEXT Bb
3440   DATA 0.8164,0.0492,-0.4831,-0.1746,0.7952,-4.5911,17.3181,-30.5563,25.287
7,-8.4570,7.6059,-36.8465,75.4833,-73.1167,26.8666,-3.8529
3450   DATA 20.6753,-44.3540,44.2607,-16.7296
3460  !
3470  SUBEND
3480  !
3490  !*************************************************************
3500  !*************************************************************
3510  !
3520  !
3530   SUB Green(Grnpos,Grn,Lr)
3540   !
3550   !THIS SUBPROGRAM EVALUATES THE VALUE OF THE GREEN'S FUNCTION
3560   !
3570   !CALC. C(ALPHA,BETA)
3580   !
3590   COM /Const/ C(*)
3600   Cab=0
3610   Alph=1/(1+Lr)
3620   FOR M=1 TO 5
3630   FOR N=0 TO 3
3640   Cab=Cab+C(M,N)*Alph^(M/2)*Grnpos^(N/2)
3650   NEXT N
3660   NEXT M
3670   !
3680   !
3690   !CALC. FBETA
```

```
3700  !
3710  Y1=Grnpos
3720  Fb=(1-Y1^2)*(.2945-.3912*Y1^2+.7635*Y1^4-.9942*Y1^6+.5094*Y1^8)
3730  !
3740  !
3750  !EVALUATE GREENS FUNCTION
3760  !
3770  Grn2=SQR(2*Y1)
3780  Grn3=2*(1+Fb)/SQR(1+Y1)-SQR(2*Y1)
3790  Grn=Grn2+Cab*Grn3
3800  IF J<Ld THEN Grn=Grn/SQR(1-Grnpos)
3810  SUBEND
3820  !
3830  !***************************************************************
3840  !***************************************************************
3850  !
3860  !
3870  SUB Integration(Lr,Radius,Integral1,Integral2,Ld,Lc1)
3880  !
3890  COM /Opt/ Opt1,Opt2
3900  COM /Zer/ Ero1(*),Ero2(*)
3910  COM /Stress/ Reg,Vals(10,20,21),R(10),Bound(10,2)
3920  COM /Cor/ Corrflag,Corrflag1
3930  Integral1=0
3940  Integral2=0
3950  !
3960  !INTEGRATION LOOP
3970  !
3980  FOR J=1 TO Ld
3990  Gauss1=0
4000  Gauss2=0
4010  FOR Lc2=1 TO 8
4020  Lc=Lc2
4030  !
4040  IF J=Ld THEN
4050  Weight=Ero1(Lc,1)
4060  Y1=Ero1(Lc,2)
4070  GOTO 4120
4080  END IF
4090  Weight=Ero2(Lc,1)
4100  Y1=Ero2(Lc,2)
4110  !
4120  Y=Y1
4130  !
4140  !OBTAIN STRESS AT CURRENT POSITION
4150  IF Opt1=0 OR Corrflag=1 THEN
4160  Cipr=Y*Lc1/Radius
4170  CALL Stress0(Cipr,Strmax,Strmin)
4180  GOTO 4260
4190  END IF
4200  IF Opt1=1 OR Opt1=2 THEN
4210  Strpos=Lc1*(Bound(J-1,0)/Lc1+(Y*(Bound(J,0)/Lc1-Bound(J-1,0)/Lc1)))
4220  IF J=Ld THEN Strpos=Lc1*(Bound(J-1,0)/Lc1+(Y*(1-Bound(J-1,0)/Lc1)))
4230  CALL Stress2(Strpos,Strmax,Strmin,Radius,J)
4240  !
4250  END IF
4260  !
4270  !EVALUATE GREENS FUNCTION AT CURRENT POSITION
4280  IF Opt1=0 OR Corrflag=1 THEN
4290  Grnpos=Y
4300  GOTO 4340
4310  END IF
4320  Grnpos=Bound(J-1,0)/Lc1+Y*(Bound(J,0)/Lc1-Bound(J-1,0)/Lc1)
4330  IF J=Ld THEN Grnpos=Bound(J-1,0)/Lc1+Y*(1-Bound(J-1,0)/Lc1)
4340  CALL Green(Grnpos,Grn,Lr)
4350  !
```

```
4360  !SUMMATION OF TERMS SO FAR
4370  IF J=Ld THEN
4380  Denom=1
4390  GOTO 4420
4400  END IF
4410  Denom=SQR((1-Bound(J-1,0)/Lc1)/(Bound(J,0)/Lc1-Bound(J-1,0)/Lc1)^Y)
4420  Gauss1=Gauss1+Strmax*Grn*Weight/Denom
4430  Gauss2=Gauss2+Strmin*Grn*Weight/Denom
4440  !
4450  !
4460  NEXT Lc2
4470  !
4480  IF J=Ld THEN
4490  Integral1=Integral1+Gauss1*SQR(1-Bound(J-1,0)/Lc1)
4500  Integral2=Integral2+Gauss2*SQR(1-Bound(J-1,0)/Lc1)
4510  GOTO 4550
4520  END IF
4530  Integral1=Integral1+Gauss1*SQR((Bound(J,0)-Bound(J-1,0))/Lc1)/2
4540  Integral2=Integral2+Gauss2*SQR((Bound(J,0)-Bound(J-1,0))/Lc1)/2
4550  NEXT J
4560  SUBEND
4570  !
4580  !
4590  !************************************************************
4600  !************************************************************
4610  !
4620  !
4630  SUB Sif1(Radius,Gauss(*),Lg)
4640  !
4650  !THIS SUBPROGRAM CALCULATES SIF'S FOR A RANGE OF CRACK LENGTHS
4660  !
4670  DIM Cla(50)
4680  COM /Stress/ Reg,Vals(*),R(*),Bound(*)
4690  COM /Opt/ Opt1,Opt2
4700  COM /Aps/ Asmax,Asmin
4710  COM /Cor/ Corrflag,Corrflag1
4720  COM /Fle/ File(*)
4730  COM /Title/ Th,B,Acc,Title$
4740  IF Opt2=1 THEN
4750  PRINT
4760  PRINT
4770  PRINT "ENTER RANGE OF CRACK."
4780  INPUT "START OF RANGE. (mm)",Str$
4790  IF LEN(Str$)=0 THEN GOTO 4770
4800  ON ERROR GOTO 4770
4810  Strt=VAL(Str$)
4820  OFF ERROR
4830  INPUT "END OF RANGE. (mm)",Fi$
4840  IF LEN(Fi$)=0 THEN GOTO 4830
4850  ON ERROR GOTO 4830
4860  Fin=VAL(Fi$)
4870  OFF ERROR
4880  IF Strt>Fin THEN GOTO 4770
4890  INPUT "INCREMENTAL LENGTH. (mm)",In$
4900  IF LEN(In$)=0 THEN GOTO 4890
4910  ON ERROR GOTO 4890
4920  Inc=VAL(In$)
4930  OFF ERROR
4940  END IF
4950  IF Opt2=2 THEN
4960  La=1
4970  LOOP
4980  Cl$=""
4990  INPUT "CRACK LENGTH? (mm)",Cl$
5000  EXIT IF LEN(Cl$)=0
5010  Cla(La)=VAL(Cl$)
```

```
5020  La=La+1
5030  END LOOP
5040  Inc=1
5050  Strt=0
5060  Fin=La-1
5070  END IF
5080  Corr=0
5090  OUTPUT 2;"K";
5100  PRINT
5110  PRINT
5120  PRINT "WHICH CORRECTION ROUTINE?"
5130  PRINT
5140  PRINT
5150  PRINT " 0. NO CORRECTION. "
5160  PRINT
5170  PRINT " 1. N. & R. WITHOUT G2."
5180  PRINT
5190  PRINT " 2. N&R/SHRIV AS CORR."
5191  Q$=""
5200  INPUT Q$
5250  ON ERROR GOTO 5191
5260  Corr=VAL(Q$)
5270  OFF ERROR
5280  IF Corr<>0 AND Corr<>1 AND Corr<>2 THEN
5290  GOTO 5191
5300  END IF
5310  OUTPUT 2;"K";
5320  IF Corr=1 OR Corr=2 THEN
5330  INPUT "THICKNESS? (mm)",Th
5340  INPUT "A/C=  ?",Aoc
5350  Corrflag1=0
5351  Q$=""
5360  INPUT "DO YOU WANT WIDTH CORRECTION?",Q$
5420  IF Q$="Y" THEN GOTO 5450
5430  IF Q$="N" THEN GOTO 5510
5440  GOTO 5351
5450  Corrflag1=1
5460  INPUT "WIDTH? (mm)",B$
5470  IF LEN(B$)=0 THEN GOTO 5460
5480  ON ERROR GOTO 5460
5490  B=VAL(B$)
5500  OFF ERROR
5510  END IF
5520  !
5530  OUTPUT 2;"K";
5540  Lp=0
5550  PRINT
5560  PRINT "CRACK LENGTH","KMAX.(MNm^-3/2)","KMIN","KMAX-KMIN","KMIN/KMAX"
5570  PRINT "        (mm)"
5580  !
5590  FOR Lcp=(Strt+Inc) TO Fin STEP Inc
5600  IF Opt2=1 THEN Lc1=Lcp
5610  IF Opt2=2 THEN Lc1=Cla(Lcp)
5620  Lc1=DROUND(Lc1,8)
5630  Lp=Lp+1
5640  IF Opt1=0 THEN
5650  Ld=1
5660  GOTO 6040
5670  END IF
5680  FOR Ld=1 TO Reg
5690  IF Lc1<=Bound(Ld,0) AND Lc1>Bound(Ld-1,0) THEN
5700  GOTO 5730
5710  END IF
5720  NEXT Ld
5730  !*************TRANSITION ZONE***************************
5740  IF Lc1<Bound(Ld-1,0)+Bound(Ld-1,0)*1.5/100 THEN
```

```
5750  L1(1)=Bound(Ld-1,0)-.000000
5760  L1(2)=Lc1+Lc1*1.5/100
5770  L1(3)=Lc1+Lc1*2/100
5780  FOR Bloop=1 TO 3
5790  Lcc=L1(Bloop)
5800  IF Bloop=1 THEN
5810  Le=Ld-1
5820  ELSE
5830  Le=Ld
5840  END IF
5850  Lr=Lcc/Radius
5860  Corrflag=0
5870  CALL Integration(Lr,Radius,Integral1,Integral2,Le,Lcc)
5880  CALL Norm(Lcc,Rg1,Rg2,Integral1,Integral2)
5890  Rg(1,Bloop)=Rg1
5900  Rg(2,Bloop)=Rg2
5910  NEXT Bloop
5920  FOR Cod=1 TO 2
5930  Dd1=(Rg(Cod,2)-Rg(Cod,1))/(L1(2)-L1(1))
5940  Dd2=(Rg(Cod,3)-Rg(Cod,2))/(L1(3)-L1(2))
5950  Dd3=(Dd2-Dd1)/(L1(3)-L1(1))
5960  P(Cod)=Rg(Cod,1)+Dd1*(Lc1-L1(1))+Dd3*(Lc1-L1(1))*(Lc1-L1(2))
5970  NEXT Cod
5980  Lr=Lc1/Radius
5990  Rg1=P(1)
6000  Rg2=P(2)
6010  GOTO 6290
6020  END IF
6030  '*****************************************************
6040  Lr=Lc1/Radius
6050  'CALCULATE INTEGRAL
6060  Corrflag=0
6070  CALL Integration(Lr,Radius,Integral1,Integral2,Ld,Lc1)
6080  '
6090  'CALCULATE NORMALISED VALUES
6100  CALL Norm(Lr,Rg1,Rg2,Integral1,Integral2)
6110  '
6120  'CORRECTION ROUTINES
6130  IF Corr=0 THEN Cf1=1
6140  IF Corr=1 AND Aoc<=1 THEN
6150  CALL Crtn1(Lc1,Cf1,Th,Radius,B,Aoc)
6160  END IF
6170  IF Corr=1 AND Aoc>1 THEN
6180  CALL Crtn3(Lc1,Cf1,Th,Radius,B,Aoc)
6190  END IF
6200  IF Corr=2 AND Aoc<=1 THEN
6210  CALL Crtn2(Lc1,Cf1,Th,Radius,B,Aoc)
6220  END IF
6230  IF Corr=2 AND Aoc>1 THEN
6240  CALL Crtn4(Lc1,Cf1,Th,Radius,B,Aoc)
6250  END IF
6260  '
6270  Rg1=Rg1*Cf1
6280  Rg2=Rg2*Cf1
6290  '
6300  'RECORD RESULTS
6310  Lg=Lp-1
6320  Gauss(Lg,1)=DROUND(Lc1,3)
6330  Gauss(Lg,2)=DROUND(Rg1*Asmax*(PI*Lc1*1.E-3)^.5,3)
6340  Gauss(Lg,3)=DROUND(Rg2*Asmax*(PI*Lc1*1.E-3)^.5,3)
6350  Gauss(Lg,4)=Gauss(Lg,2)-Gauss(Lg,3)
6360  Gauss(Lg,5)=DROUND(Rg2/Rg1,3)
6370  File(1,Lg,1)=Gauss(Lg,1)
6380  File(1,Lg,2)=Gauss(Lg,2)
6390  '
6400  PRINT "  ";Gauss(Lg,1),"",Gauss(Lg,2),"",Gauss(Lg,3),Gauss(Lg,4),Gauss(Lg
```

```
,5)
6410  NEXT Lcp
6420  File(1,0,0)=Lcp-2
6430  !
6440  !
6450  SUBEND
6460  !
6470  !*************************************************************
6480  !*************************************************************
6490  !
6500  !
6510  SUB Stress2(Strpos,Strmax,Strmin,Radius,J)
6520  Cip=Strpos
6530  ! CALC. STRESS ACCORDING TO INTERPOLATION
6540  !
6550  COM /Stress/ Reg,Vals(*),R(*),Bound(*)
6560  COM /Aps/ Asmax,Asmin
6570  Y=J-1
6580  Interp=Vals(Y,1,R(Y))
6590  FOR X=1 TO (R(Y)-1)
6600  E=R(Y)-X
6610  Interp=Interp*(Cip-Vals(Y,E,0))+Vals(Y,1,E)
6620  NEXT X
6630  Es=Interp
6640  IF Strpos>Bound(Reg,0) THEN Es=0
6650  !
6660  Cipr=Cip/Radius
6670  !
6680  !CALC. APPLIED STRESS
6690  !
6700  Aps=.5*(2+(1/(1+Cipr))^2+3*(1/(1+Cipr))^4)
6710  !
6720  Strmax=Aps+Es
6730  Strmin=(Aps*Asmin/Asmax)+Es
6740  !
6750  SUBEND
6760  !
6770  !*************************************************************
6780  !*************************************************************
6790  !
6800  SUB Norm(Lc1,Rg1,Rg2,Integral1,Integral2)
6810  !
6820  COM /Aps/ Asmax,Asmin
6830  Rg1=Integral1/PI
6840  Rg2=Integral2/PI
6850  !
6860  !
6870  SUBEND
6880  !
6890  !*************************************************************
6900  !*************************************************************
6910  !
6920  !
6930  SUB Stress0(Cipr,Strmax,Strmin)
6940  !
6950  COM /Aps/ Asmax,Asmin
6960  Aps=.5*(2+(1/(1+Cipr))^2+3*(1/(1+Cipr))^4)
6970  !
6980  Strmax=Aps
6990  Strmin=Aps*Asmin/Asmax
7000  !
7010  SUBEND
7020  !
7030  !*************************************************************
7040  !*************************************************************
7050  !
```

```
7060  !
7070  SUB Plt(Ult,Zz(*),Str$,Xlabel$,Ylabel$)
7080  GINIT
7090  GCLEAR
7100  GRAPHICS ON
7110  Xmax=Zz(1,1)
7120  Xmin=Zz(1,1)
7130  Ymax=Zz(1,2)
7140  Ymin=Zz(1,2)
7150  FOR Hh=0 TO Ult
7160  IF Zz(Hh,1)>Xmax THEN Xmax=Zz(Hh,1)
7170  IF Zz(Hh,1)<Xmin THEN Xmin=Zz(Hh,1)
7180  IF Zz(Hh,2)>Ymax THEN Ymax=Zz(Hh,2)
7190  IF Zz(Hh,2)<Ymin THEN Ymin=Zz(Hh,2)
7200  NEXT Hh
7210  PRINT
7220  PRINT
7230  PRINT "X.RANGE= ";Xmin;" TO ";Xmax
7240  PRINT "Y.RANGE= ";Ymin;" TO ";Ymax
7250  INPUT "X-AXIS LABEL SPACING?",Xsp$
7260  IF LEN(Xsp$)=0 THEN GOTO 7250
7270  ON ERROR GOTO 7250
7280  Xspc=VAL(Xsp$)
7290  OFF ERROR
7300  INPUT "Y-AXIS LABEL SPACING?",Ysp$
7310  IF LEN(Xsp$)=0 THEN GOTO 7300
7320  ON ERROR GOTO 7300
7330  Yspc=VAL(Ysp$)
7340  OFF ERROR
7350  OUTPUT 2;"K";
7360  MOVE 35,95
7370  LABEL Str$
7380  VIEWPORT 15,115,30,90
7390  FRAME
7400  Xdiff=Xmax-Xmin
7410  Ydiff=Ymax-Ymin
7420  WINDOW Xmin,Xmax+Xdiff/10,Ymin-Ydiff/10,Ymax+Ydiff/10
7430  AXES Xspc/5,Yspc/5,0,0,5,5,1
7440  CLIP OFF
7450  CSIZE 3
7460  FOR Xlab=0 TO Xmax+Xdiff/12 STEP Xspc
7470  IF Xlab<Xmin-Xdiff/12 THEN GOTO 7500
7480  MOVE Xlab-Xdiff/140,Ymin-2*Ydiff/10
7490  LABEL USING "#,K";Xlab
7500  NEXT Xlab
7510  MOVE Xmax+Xdiff/8,Ymin-Ydiff/5
7520  LABEL USING "#,K";Xlabel$
7530  FOR Ylab=0 TO Ymax+Ydiff/12 STEP Yspc
7540  IF Ylab<Ymin-Ydiff/12 THEN GOTO 7570
7550  MOVE Xmin-Xdiff/10,Ylab-Ydiff/30
7560  LABEL USING "#,K";Ylab
7570  NEXT Ylab
7580  FOR Ylab=0 TO Ymin-Ydiff/10 STEP -Yspc
7590  MOVE Xmin-Xdiff/10,Ylab-Ydiff/30
7600  LABEL USING "#,K";Ylab
7610  NEXT Ylab
7620  MOVE Xmin-Xdiff/10,Ymax+Ydiff/8
7630  LABEL USING "#,K";Ylabel$
7640  P=1
7650  FOR Rr=0 TO Ult
7660  PLOT Zz(Rr,1),Zz(Rr,2),P
7670  NEXT Rr
7671  Q$=""
7680  INPUT "COPY?",Q$
7740  IF Q$="Y" THEN GOTO 7780
7750  IF Q$="N" THEN GOTO 7800
```

```
7760   GOTO 7671
7770   !
7780   OUTPUT 2;"N";
7790   !
7800   SUBEND
7810   !
7820   !*************************************************************
7830   !*************************************************************
7840   !
7850   SUB Inputs(Radius)
7860   !
7870   COM /Opt/ Opt1,Opt2
7880   COM /Stress/ Reg,Vals(*),R(*),Bound(*)
7890   COM /Aps/ Asmax,Asmin
7900   COM /Fle/ File(*)
7910   COM /Title/ Th,B,Aoc,Name$
7920   DIM Lz(50,2),Zz(102,2)
7930   PRINT
7940   !
7950   IF Opt1=2 THEN
7960   PRINT "RSF LOADED FROM FILE"
7970   GOTO 8000
7980   PRINT "FILE NOT FOUND."
7990   OFF ERROR
8000   INPUT "NAME OF FILE?",Name$
8010   IF LEN(Name$)=0 THEN GOTO 8000
8020   OUTPUT 2;"K";
8030   PRINT
8040   PRINT
8050   ON ERROR GOTO 7980
8060   ASSIGN @Path TO Name$
8070   OFF ERROR
8080   ON ERROR GOTO 8100
8090   ENTER @Path;Lz(*)
8100   OFF ERROR
8110   ASSIGN @Path TO *
8120   PRINT "FILE:            ",Name$
8130   PRINT "NO. OF ENTRIES:",Lz(2,0)
8140   PRINT "RADIUS OF HOLE:",Lz(3,0)
8150   Radius=Lz(3,0)
8160   Ult=Lz(2,0)-1
8170   !
8180   !
8190   ELSE
8200   Name$="KEYBOARD ENTRY."
8210   OUTPUT 2;"K";
8220   INPUT "RADIUS? (mm)",Radi$
8230   IF LEN(Radi$)=0 THEN GOTO 8220
8240   ON ERROR GOTO 8220
8250   Radius=VAL(Radi$)
8260   OFF ERROR
8261   PRINT "ORIGIN OF DISPLACEMENT COORDS."
8263   Q$=""
8270   INPUT "0 FOR CENTRE OF HOLE, 1 FOR EDGE.",Q$
8330   ON ERROR GOTO 8263
8340   Kd=VAL(Q$)
8350   OFF ERROR
8360   IF Kd<>0 AND Kd<>1 THEN GOTO 8263
8370   Lz(1,0)=Kd
8380   !
8390   Lz(3,0)=Radius
8400   PRINT "INPUT POSITIONS AND STRESS VALUES."
8410   PRINT
8420   PRINT
8430   PRINT "NO.","X.","STRESS."
8440   Inp=-1
```

```
8450   LOOP
8460   Inp=Inp+1
8470   Valx$=""
8480   INPUT "VALUE OF X ? (mm)",Valx$
8490   EXIT IF LEN(Valx$)=0
8500   ON ERROR GOTO 8480
8510   Lz(Inp,1)=VAL(Valx$)
8520   OFF ERROR
8530   INPUT "VALUE OF STRESS? (MPa)",Vals$
8540   IF LEN(Vals$)=0 THEN GOTO 8530
8550   ON ERROR GOTO 8530
8560   Valstr=VAL(Vals$)
8570   OFF ERROR
8580   Lz(Inp,2)=Valstr
8590   PRINT Inp,Lz(Inp,1),Lz(Inp,2)
8600   END LOOP
8601   Q$=""
8610   INPUT "CHANGE ANY?",Q$
8670   IF Q$="Y" THEN GOTO 8710
8680   IF Q$="N" THEN GOTO 8760
8690   GOTO 8601
8700   !
8710   INPUT "ENTRY TO BE CHANGED?",Nm
8720   INPUT "VALUE OF X. (mm)",Lz(Nm,1)
8730   INPUT "VALUE OF STRESS. (MPa)",Lz(Nm,2)
8740   PRINT TABXY(1,Nm+6),Nm,Lz(Nm,1),Lz(Nm,2)
8750   GOTO 8601
8760   Ult=Inp-1
8770   Lz(2,0)=Inp
8780   !
8781   Q$=""
8790   INPUT "STORE DISTRIBUTION?",Q$
8850   IF Q$="Y" THEN GOTO 8881
8860   IF Q$="N" THEN GOTO 9190
8870   GOTO 8781
8880   !
8881   Q$=""
8890   INPUT "CREATE NEW FILE?",Q$
8950   IF Q$="Y" THEN GOTO 8990
8960   IF Q$="N" THEN GOTO 9100
8970   GOTO 8900
8980   !
8990   GOTO 9010
9000   PRINT TABXY(4,10);"FILENAME ALREADY USED. TRY ANOTHER."
9010   INPUT "FILE NAME?",Name$
9020   IF LEN(Name$)=0 THEN GOTO 9010
9030   ON ERROR GOTO 9000
9040   CREATE BDAT Name$,5
9050   OFF ERROR
9060   GOTO 9120
9070   OUTPUT 2;"K";
9080   PRINT TABXY(4,10);"FILE NOT FOUND. TRY AGAIN."
9090   PRINT
9100   INPUT "FILE NAME?",Name$
9110   ON ERROR GOTO 9080
9120   ASSIGN @Pth TO Name$
9130   OFF ERROR
9140   OUTPUT @Pth;Lz(*)
9150   ASSIGN @Pth TO *
9160   !
9170   END IF
9180   GOTO 9234
9190   !
9200   INPUT "TITLE?",Name$
9201   Q$=""
9210   INPUT "COPY OF COORDS?",Q$
```

```
9211   IF Q$="Y" THEN
9212   PRINTER IS 701
9213   PRINT Name$
9214   PRINT "RADIUS=";Radius;" mm."
9215   IF Lz(1,0)=0 THEN
9216   PRINT "L MEASURED FROM CENTRE OF HOLE."
9217   END IF
9218   IF Lz(1,0)=1 THEN
9219   PRINT "L MEASURED FROM EDGE OF HOLE."
9220   END IF
9222   PRINT
9223   PRINT "L (mm)","STRESS (MPa)"
9224   FOR Po=0 TO Ult
9225   PRINT Lz(Po,1),Lz(Po,2)
9226   NEXT Po
9227   PRINTER IS 1
9228   GOTO 9234
9230   END IF
9231   IF Q$="N" THEN GOTO 9234
9232   GOTO 9201
9233   !
9234   IF Lz(1,0)=0 THEN
9240   FOR Yy=0 TO 50
9250   Lz(Yy,1)=Lz(Yy,1)-Radius
9260   NEXT Yy
9270   END IF
9280   CALL Plt(Ult,Lz(*),Name$,"X","STR")
9290   INPUT "'NO. OF REGIONS?",Re$
9300   IF LEN(Re$)=0 THEN GOTO 9290
9310   ON ERROR GOTO 9290
9320   Reg=VAL(Re$)
9330   OFF ERROR
9340   GCLEAR
9350   FOR Z=0 TO Ult
9360   PRINT Z,Lz(Z,1),Lz(Z,2)
9370   File(0,Z,1)=Lz(Z,1)
9380   File(0,Z,2)=Lz(Z,2)
9390   NEXT Z
9400   File(0,0,0)=Ult
9410   Boun(0)=0
9411   Q$=""
9413   INPUT "COPY.",Q$
9414   IF Q$="Y" THEN
9415   PRINTER IS 701
9416   PRINT Name$
9417   PRINT
9419   PRINT "L (mm)","STRESS (MPa)"
9420   FOR Pp=0 TO Ult
9421   PRINT Lz(Pp,1),Lz(Pp,2)
9422   NEXT Pp
9423   PRINTER IS 1
9424   GOTO 9428
9425   END IF
9426   IF Q$="N" THEN GOTO 9428
9427   GOTO 9411
9428   PRINT "INPUT NUMBERS OF POSITIONS OF THE BOUNDARIES."
9430   PRINT
9440   PRINT
9450   PRINT
9460   FOR X=1 TO Reg
9470   PRINT USING "+,K";"POSITION OF END OF REGION",X
9480   INPUT Boun(X)
9490   Bound(X,0)=Lz(Boun(X),1)
9500   NEXT X
9510   !
9520   Bound(0,0)=Lz(0,1)
```

```
9530   FOR Bnds=0 TO Reg-1
9540   Lpn=0
9550   FOR Pts=Boun(Bnds) TO Boun(Bnds+1)
9560   Lpn=Lpn+1
9570   Vals(Bnds,Lpn,0)=Lz(Pts,1)
9580   Vals(Bnds,Lpn,1)=Lz(Pts,2)/Asmax
9590   NEXT Pts
9600   R(Bnds)=Lpn
9610   NEXT Bnds
9620   !
9630   !CALC. DIVIDED DIFF. TABLE
9640   FOR Z1=0 TO (Reg-1)
9650   FOR X1=2 TO (R(Z1)+1)
9660   FOR Y1=1 TO (R(Z1)-(X1-1))
9670   Ii=(Vals(Z1,Y1+1,X1-1)-Vals(Z1,Y1,X1-1))/(Vals(Z1,Y1+X1-1,0)-Vals(Z1,Y1,0
))
9680   Vals(Z1,Y1,X1)=DROUND(Ii,5)
9690   NEXT Y1
9700   NEXT X1
9710   NEXT Z1
9720   !
9730   !PLOT ROUTINE
9740   !
9750   Ult=0
9760   Yy=0
9770   Xmin=Bound(0,0)
9780   Xmax=Bound(*-1,0)
9790   Xdiff=Xmax-Xmin
9800   FOR Range=Xmin TO Xmax STEP (Xdiff/100)
9810   Ult=Ult+1
9820   IF Range>Bound(Yy+1,0) THEN Yy=Yy+1
9830   Interp=Vals(Yy,1,R(Yy))
9840   !
9850   FOR Xx=1 TO (R(Yy)-1)
9850   E=R(Yy)-Xx
9870   Interp=Interp*(Range-Vals(Yy,E,0))+Vals(Yy,1,E)
9880   NEXT Xx
9890   Zz(Ult,1)=Range
9900   Zz(Ult,2)=Interp*Asmax
9910   NEXT Range
9920   DIM Str$[30]
9930   Str$="RESIDUAL STRESS FIELD"
9940   CALL Plt(Ult,Zz(*),Str$,"L","STR")
9950   !
9951   Q$=""
9960   INPUT "OK?",Q$
10020  IF Q$="Y" THEN GOTO 10090
10030  IF Q$="N" THEN
10040  Ult=Lz(2,0)-1
10050  GOTO 9280
10060  END IF
10070  GOTO 9951
10080  !
10090  SUBEND
10100  !
10110  !*************************************************************************
10120  !*************************************************************************
10130  !
10140 SUB Rate(U,W,R2,K2)
10150!*********************************************************
10160!***********CRACK RATE DATA : NAME,KCRIT,DATA***********
10170!******************(ALL IN N/mm^3/2)******************
10180!*********************************************************
10190 DATA L71
10200 Kcritd=1.86E+3
10210 K1=Kcritd
```

```
10220 DATA -2,98.9,102,115,147,315,950,1800,-1,1E-7,6E-7,1.5E-6,4E-6,4E-5,1E-3,1
E-2,0
10230 DATA -1,65.9,69.0,80.0,101,495,1120,-1,1E-7,4E-7,1E-6,3E-6,4E-4,1E-2,0
10240 DATA -.5,49.5,51.0,65.0,450,750,-1,1E-7,4E-7,1.5E-6,1E-3,1E-2,0
10250 DATA -.25,41.3,43.0,50.0,450,620,-1,1E-7,4E-7,9E-7,2E-3,1E-2,0
10260 DATA 0,33.0,35.0,40.0,390,510,-1,1E-7,3E-7,7.3E-7,2E-3,1E-2,0
10270 DATA .25,26.6,28.0,32.0,235,390,-1,1E-7,2E-7,4E-7,6E-4,1E-2,0
10280 DATA .5,23.4,24.5,28.0,149,300,-1,1E-7,2E-7,4E-7,1.55E-4,1E-2,0
10290 DATA 999
10300!*****************************************************************
10310 Opdev=701
10320 Data$="YES"
10330!************************************************
10340!***************READING IN OF DATA***************
10350!************************************************
10360!**********************
10370!****CRACK RATE DATA****
10380!**********************
10390 READ Ratename$
10400 I1=0
10410 LOOP
10420 READ Ts
10430 EXIT IF Ts=999
10440 R(I1)=Ts
10450 I2=0
10460 LOOP
10470 READ Dk
10480 EXIT IF Dk=-1.E+0
10490 Ak(I1,I2)=LOG(Dk)
10500 I2=I2+1
10510 END LOOP
10520 I3=0
10530 LOOP
10540 READ Rte
10550 EXIT IF Rte=0
10560 S(I1,I3)=LOG(Rte)
10570 I3=I3+1
10580 END LOOP
10590 I1=I1+1
10600 END LOOP
10610 I=I1-1
10620!
10630 IF U+W>K2 THEN GOTO 10850
10640 IF W<=0 THEN GOTO 10830
10650 IF U+W<=0 THEN GOTO 10830
10660 R3=(U-W)/(U+W)
10670 P=U+W
10680 IF R3>R(0) THEN GOTO 10870
10690 R3=R(0)
10700 W1=LOG(P/2*(1-R3))
10710 J=0
10720 IF W1<Ak(J,0) THEN GOTO 10830
10730 FOR K=1 TO 9
10740 IF Ak(J,K)>W1 THEN GOTO 10770
10750 IF Ak(J,K+1)=0 THEN GOTO 10770
10760 NEXT K
10770 K3=W1-Ak(J,K)
10780 Z=S(J,K)+K3*(S(J,K)-S(J,K-1))/(Ak(J,K)-Ak(J,K-1))
10790 IF K3<0 THEN GOTO 10810
10800 Z=Z+K3*K3/((LOG(K1*(1-R3)/2)-W1+K3)^2-K3*K3)
10810 R2=EXP(Z)*((1-P/K1)/(1-(U+W)/K2))^.5
10820 GOTO 11250
10830 R2=0
10840 GOTO 11250
10850 R2=1.E+10
10860 GOTO 11250
```

```
10870 W1=LOG(W)
10880 IF R(I)>R3 THEN GOTO 10930
10890 R3=R(I)
10900 J=I
10910 P=2*W/(1-R3)
10920 GOTO 10720
10930 FOR J=1 TO I
10940 IF R(J)>R3 THEN GOTO 10960
10950 NEXT J
10960 F=(R3-R(J-1))/(R(J)-R(J-1))
10970 H3=F*Ak(J,0)+(1-F)*Ak(J-1,0)
10980 IF W1<H3 THEN GOTO 10830
10990 K=0
11000 L=0
11010 H4=S(J,0)
11020 IF S(J,K+1)>S(J-1,L+1) THEN GOTO 11140
11030 K=K+1
11040 IF S(J,K)=S(J-1,L) THEN GOTO 11020
11050 H1=S(J,K)
11060 G=(S(J-1,L+1)-S(J-1,L))/(Ak(J-1,L+1)-Ak(J-1,L))
11070 H2=Ak(J-1,L)+(S(J,K)-S(J-1,L))/G
11080 H5=F*Ak(J,K)+(1-F)*H2
11090 IF Ak(J,K+1)=0 THEN GOTO 11220
11100 IF W1<H5 THEN GOTO 11220
11110 H3=H5
11120 H4=H1
11130 GOTO 11020
11140 L=L+1
11150 IF S(J,K)=S(J-1,L) THEN GOTO 11020
11160 H1=S(J-1,L)
11170 G=(S(J,K+1)-S(J,K))/(Ak(J,K+1)-Ak(J,K))
11180 H2=Ak(J,K)+(S(J-1,L)-S(J,K))/G
11190 H5=F*H2+(1-F)*Ak(J-1,L)
11200 IF Ak(J-1,L+1)=0 THEN GOTO 11220
11210 GOTO 11100
11220 K3=W1-H5
11230 Z=H1+K3*(H1-H4)/(H5-H3)
11240 GOTO 10790
11250 R2=DROUND(R2,3)
11260 SUBEND
11270 !
11280 !***********************************************************
11290 !***********************************************************
11300 !
11310 SUB Crtn1(C,Fch,T,Radius,B,Aoc)
11320 COM /Cor/ Corrflag,Corrflag1
11330 !
11340 Thi=0
11350 A=C*Aoc
11360 Q=1+1.464*(A/C)^1.65
11370 M1=1.13-.09*(A/C)
11380 M2=-.54+.89/(.2+(A/C))
11390 M3=.5-1/(.65+(A/C))+14*(1-A/C)^24
11400 G1=1+(.1+.35*(A/T)^2)*(1-SIN(Thi))^2
11410 G3=(1+.04*(A/C))*(1+.1*(1-COS(Thi))^2)*(.85+.15*(A/T)^.25)
11420 Fthi=((A/C)^2*COS(Thi)^2+SIN(Thi)^2)^.25
11430 IF Corrflag1=1 THEN
11440 Fw=(1/COS(PI*Radius/(2*B))*1/COS(PI*(2*Radius+C)/(4*(B-C)+2*C)*SQR(A/T)))^
.5
11450 ELSE
11460 Fw=1
11470 END IF
11480 Fch=((M1+M2*(A/T)^2+M3*(A/T)^4)*G1*G3*Fthi*Fw)/SQR(Q)
11490 K1=SQR((4/PI+A*C/(2*T*Radius))/(4/PI+A*C/(T*Radius)))
11500 Fch=Fch*K1
11510 SUBEND
```

```
11520 !
11530 !
11540 !*************************************************************
11550 !*************************************************************
11560 !
11570 SUB Crtn2(C,Cf1,T,Radius,B,Aoc)
11580 !
11590 !
11600 COM /Cor/ Corrflag,Corrflag1
11610 Thi=0
11620 A=C*Aoc
11630 Q=1+1.464*(A/C)^1.65
11640 M1=1.13-.09*(A/C)
11650 M2=-.54+.89/(.2+(A/C))
11660 M3=.5-1/(.65+(A/C))+14*(A/C)^24
11670 G1=1+(.1+.35*(A/T)^2)*(1-SIN(Thi))^2
11680 Lam=1/(1+(C/Radius)*COS(.85*Thi))
11690 G2=(1+.358*Lam+1.425*Lam^2-1.578*Lam^3+2.156*Lam^4)/(1+.13*Lam^2)
11700 G3=(1+.04*A/C)*(1+.1*(1-COS(Thi))^2)*(.85+.15*(A/T)^.25)
11710 Fthi=((A/C)^2*COS(Thi)^2+SIN(Thi)^2)^.25
11720 IF Corrflag1=1 THEN
11730 Fw=(1/COS(PI*Radius/(2*B))*1/COS(PI*(2*Radius+C)/(4*(B-C)+2*C)*SQR(A/T)))^
.5
11740 ELSE
11750 Fw=1
11760 END IF
11770 Nr=((M1+M2*(A/T)^2+M3*(A/T)^4)*G1*G2*G3*Fthi*Fw)/SQR(Q)
11780 K1=SQR((4/PI+A*C/(2*T*Radius))/(4/PI+A*C/(T*Radius)))
11790 Nr=Nr*K1
11800 !
11810 !
11820 Lr=C/Radius
11830 Corrflag=1
11840 CALL Integration(Lr,Radius,Integral1,Integral2,1,C)
11850 CALL Norm(Lr,Rg1,Rg2,Integral1,Integral2)
11860 Shiv1=Rg1
11870 Cf1=Nr/Shiv1
11880 !
11890 SUBEND
11900 !
11910 !**************************************************
11920 !**************************************************
11930 SUB Crtn3(C,Fch,T,Radius,B,Aoc)
11940 COM /Cor/ Corrflag,Corrflag1
11950 !
11960 Thi=0
11970 A=C*Aoc
11980 Q=1+1.464*(C/A)^1.65
11990 M1=SQR(C/A)*(1+.04*(C/A))
12000 M2=.2*(C/A)^4
12010 M3=-.11*(C/A)^4
12020 G1=1+(.1+.35*(C/A)*(A/T)^2)*(1-SIN(Thi))^2
12030 G3=(1.13-.09*(C/A))*(1+.1*(1-COS(Thi))^2)*(.85+.15*(A/T)^.25)
12040 Fthi=((C/A)^2*SIN(Thi)^2+COS(Thi)^2)^.25
12050 IF Corrflag1=1 THEN
12060 Fw=(1/COS(PI*Radius/(2*B))*1/COS(PI*(2*Radius+C)/(4*(B-C)+2*C)*SQR(A/T)))^
.5
12070 ELSE
12080 Fw=1
12090 END IF
12100 Fch=((M1+M2*(A/T)^2+M3*(A/T)^4)*G1*G3*Fthi*Fw)/SQR(Q)
12110 K1=SQR((4/PI+A*C/(2*T*Radius))/(4/PI+A*C/(T*Radius)))
12120 Fch=Fch*K1
12130 SUBEND
12140 !
12150 !**************************************************************
```

```
12160 !*****************************************************************
12170 !
12180 SUB Crtn4(C,Cf1,T,Radius,B,Aoc)
12190 COM /Cor/ Corrflag,Corrflag1
12200 Thi=0
12210 A=C*Aoc
12220 Q=1+1.464*(C/A)^1.65
12230 M1=SQR(C/A)*(1+.04*(C/A))
12240 M2=.2*(C/A)^4
12250 M3=-.11*(C/A)^4
12260 G1=1+(.1+.35*(C/A)*(A/T)^2)*(1-SIN(Thi))^2
12270 Lam=1/(1+(C/Radius)*COS(.85*Thi))
12280 G2=(1+.358*Lam+1.425*Lam^2-1.578*Lam^3+2.156*Lam^4)/(1+.13*Lam^2)
12290 G3=(1.13-.09*C/A)*(1+.1*(1-COS(Thi))^2)*(.85+.15*(A/T)^.25)
12300 Fthi=((C/A)^2*SIN(Thi)^2+COS(Thi)^2)^.25
12310 IF Corrflag1=1 THEN
12320 Fw=(1/COS(PI*Radius/(2*B))*1/COS(PI*(2*Radius+C)/(4*(B-C)+2*C)*SQR(A/T)))^
.5
12330 ELSE
12340 Fw=1
12350 END IF
12360 Nr=((M1+M2*(A/T)^2+M3*(A/T)^4)*G1*G2*G3*Fthi*Fw)/SQR(Q)
12370 K1=SQR((4/PI+A*C/(2*T*Radius))/(4/PI+A*C/(T*Radius)))
12380 Nr=Nr*K1
12390 !
12400 !
12410 Lr=C/Radius
12420 Corrflag=1
12430 CALL Integration(Lr,Radius,Integral1,Integral2,1,C)
12440 CALL Norm(Lr,Rg1,Rg2,Integral1,Integral2)
12450 Shiv1=Rg1
12460 Cf1=Nr/Shiv1
12470 !
12480 SUBEND
12490 !
12500 !**************************************************************
12510 !**************************************************************
12520 !
12530 !
12540 SUB Mplot
12550 !
12560 DIM Mplt(5,2,50,2)
12570 DIM Hld(2,50,2)
12580 DIM Pld(5,50,2)
12590 !
12600 GCLEAR
12610 OUTPUT 2;"K";
12620 !
12630 Nm=0
12640 !
12650 LOOP
12660 Nm=Nm+1
12670 OUTPUT 2;"K";
12680 GOTO 12710
12690 OUTPUT 2;"K";
12700 PRINT "FILE NOT FOUND."
12710 INPUT "NAME OF FILE?",Name$
12720 IF LEN(Name$)=0 THEN GOTO 12710
12730 ON ERROR GOTO 12700
12740 ASSIGN @Path1 TO Name$
12750 OFF ERROR
12760 ENTER @Path1;Hld(*)
12770 ASSIGN @Path1 TO *
12780 FOR X=0 TO 2
12790 FOR Y=0 TO 50
12800 FOR Z=0 TO 2
```

```
12810 Mplt(Nm,X,Y,Z)=Hld(X,Y,Z)
12820 NEXT Z
12830 NEXT Y
12840 NEXT X
12850 !
12860 !
12861 Q$=""
12870 INPUT "ANOTHER FILE?",Q$
12930 IF Q$="Y" THEN GOTO 12970
12940 EXIT IF Q$="N"
12950 GOTO 12861
12960 !
12970 END LOOP
12980 !
12990 PRINT "0.   EXIT."
13000 PRINT "1.   RESIDUAL STRESS FIELD."
13010 PRINT "2.   STRESS INTENSITY FACTOR DIST."
13020 PRINT "3.   CRACK GROWTH RATE DIST."
13030 Nd=1
13031 Q$=""
13040 INPUT "WHICH DO YOU WANT TO PLOT?",Q$
13100 ON ERROR GOTO 13031
13110 Nd=VAL(Q$)
13120 OFF ERROR
13130 IF Nd<>0 AND Nd<>1 AND Nd<>2 AND Nd<>3 THEN
13140 GOTO 13031
13150 END IF
13160 IF Nd=0 THEN GOTO 13450
13170 FOR Md=1 TO Nm
13180 FOR Xz=0 TO 50
13190 Pld(Md,Xz,1)=Mplt(Md,Nd-1,Xz,1)
13200 Pld(Md,Xz,2)=Mplt(Md,Nd-1,Xz,2)
13210 Pld(Md,0,0)=Mplt(Md,Nd-1,0,0)
13220 NEXT Xz
13230 NEXT Md
13240 DIM Lab$[30]
13250 IF Nd=1 THEN
13260 X$="X"
13270 Y$="CTR"
13280 Lab$="RESIDUAL STRESS DIST."
13290 END IF
13300 IF Nd=2 THEN
13310 X$="X"
13320 Y$="SIF"
13330 Lab$="STRESS INT FACTOR DIST."
13340 END IF
13350 IF Nd=3 THEN
13360 X$="X"
13370 Y$="RATE"
13380 Lab$="CRACK GROWTH RATES."
13390 END IF
13400 CALL Plt1(Pld(*),Lab$,X$,Y$,Nm)
13410 !
13420 GCLEAR
13430 OUTPUT 2;"K";
13440 GOTO 12990
13450 SUBEND
13460 !
13470 !*****************************************************
13480 !*****************************************************
13490 !
13500  SUB Plt1(Pts(*),Str$,Xlabel$,Ylabel$,Nm)
13510  GINIT
13520  GRAPHICS ON
13530  Xmax=Pts(1,1,1)
13540  Xmin=Pts(1,1,1)
```

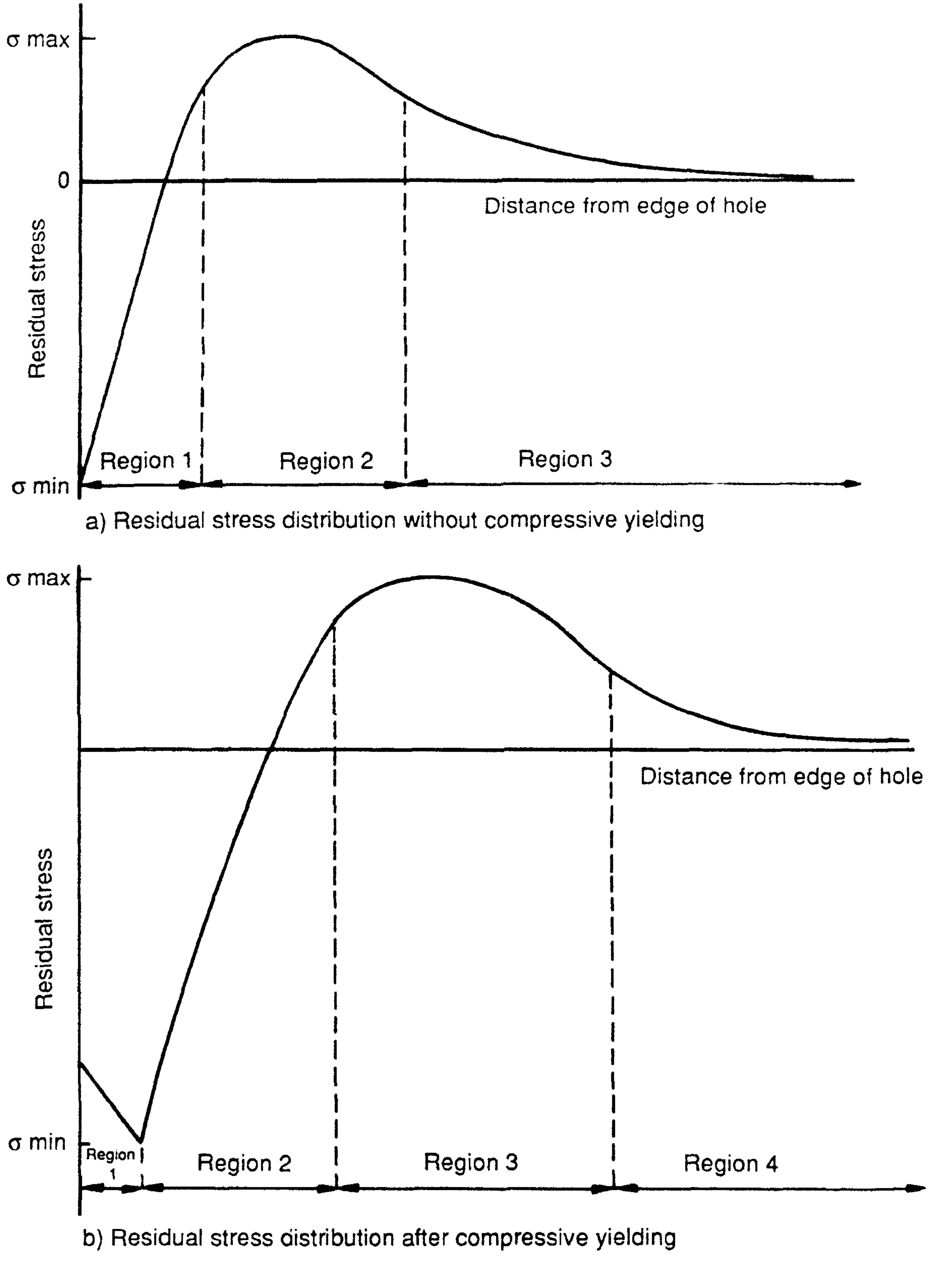
```
13550  Ymax=Pts(1,1,2)
13560  Ymin=Pts(1,1,2)
13570  FOR Gg=1 TO Nm
13580  FOR Hh=0 TO Pts(Gg,0,0)+1
13590  IF Pts(Gg,Hh,1)>Xmax THEN Xmax=Pts(Gg,Hh,1)
13600  IF Pts(Gg,Hh,1)<Xmin THEN Xmin=Pts(Gg,Hh,1)
13610  IF Pts(Gg,Hh,2)>Ymax THEN Ymax=Pts(Gg,Hh,2)
13620  IF Pts(Gg,Hh,2)<Ymin THEN Ymin=Pts(Gg,Hh,2)
13630  NEXT Hh
13640  NEXT Gg
13650  PRINT
13660  PRINT
13670  PRINT "X.RANGE= ";Xmin;" TO ";Xmax
13680  PRINT "Y.RANGE= ";Ymin;" TO ";Ymax
13690  INPUT "X-AXIS LABEL SPACING?",Xsp$
13700  IF LEN(Xsp$)=0 THEN GOTO 13690
13710  ON ERROR GOTO 13690
13720  Xspc=VAL(Xsp$)
13730  OFF ERROR
13740  INPUT "Y-AXIS LABEL SPACING?",Ysp$
13750  IF LEN(Ysp$)=0 THEN GOTO 13740
13760  ON ERROR GOTO 13740
13770  Yspc=VAL(Ysp$)
13780  OFF ERROR
13790  OUTPUT 2;"K";
13800  MOVE 35,95
13810  LABEL Str$
13820  VIEWPORT 15,115,30,90
13830  FRAME
13840  Xdiff=Xmax-Xmin
13850  Ydiff=Ymax-Ymin
13860  WINDOW Xmin,Xmax+Xdiff/10,Ymin-Ydiff/10,Ymax+Ydiff/10
13870  AXES Xspc/5,Yspc/5,0,0,5,5,1
13880  CLIP OFF
13890  CSIZE 3
13900  FOR Xlab=0 TO Xmax+Xdiff/12 STEP Xspc
13910  IF Xlab<Xmin-Xdiff/12 THEN GOTO 13940
13920  MOVE Xlab-Xdiff/140,Ymin-2*Ydiff/10
13930  LABEL USING "#,K";Xlab
13940  NEXT Xlab
13950  MOVE Xmax+Xdiff/8,Ymin-Ydiff/5
13960  LABEL USING "#,K";Xlabel$
13970  FOR Ylab=0 TO Ymax+Ydiff/12 STEP Yspc
13980  IF Ylab<Ymin-Ydiff/12 THEN GOTO 14010
13990  MOVE Xmin-Xdiff/10,Ylab-Ydiff/30
14000  LABEL USING "#,K";Ylab
14010  NEXT Ylab
14020  FOR Ylab=0 TO Ymin-Ydiff/10 STEP -Yspc
14030  MOVE Xmin-Xdiff/10,Ylab-Ydiff/30
14040  LABEL USING "#,K";Ylab
14050  NEXT Ylab
14060  MOVE Xmin-Xdiff/10,Ymax+Ydiff/8
14070  LABEL USING "#,K";Ylabel$
14080  P=1
14090  FOR Qq=1 TO Nm
14100  LINE TYPE Qq+3
14110  FOR Rr=0 TO Pts(Qq,0,0)
14120  PLOT Pts(Qq,Rr,1),Pts(Qq,Rr,2),P
14130  NEXT Rr
14140  MOVE 0,0
14150  NEXT Qq
14151  Q$=""
14160  INPUT "COPY?",Q$
14170  IF Q$="Y" THEN GOTO 14260
1      IF Q$="N" THEN GOTO 14280
1      GOTO 14151
```

```
14250  !
14260  OUTPUT 2;"N";
14270  !
14280  SUBEND
14290  !
14300  !**************************************************************
14310  !**************************************************************
14320  !
14330 SUB Dplot(Gauss(*),Ult)
14331 !DELTA K AND R
14332 !
14338 Xmax=Gauss(0,1)
14339 Xmin=Gauss(0,1)
14340 Y1max=Gauss(0,4)
14341 Y2min=Gauss(0,5)
14344 FOR Pp=0 TO Ult
14345 IF Gauss(Pp,1)>Xmax THEN Xmax=Gauss(Pp,1)
14346 IF Gauss(Pp,1)<Xmin THEN Xmin=Gauss(Pp,1)
14347 IF Gauss(Pp,4)>Y1max THEN Y1max=Gauss(Pp,4)
14348 IF Gauss(Pp,5)<Y2min THEN Y2min=Gauss(Pp,5)
14351 NEXT Pp
14353 PRINT "XRANGE=  0 TO";Xmax
14354 INPUT "X-AXIS LABEL SPACING?",Xspc
14362 !
14363 OUTPUT 2;"K";
14365 GINIT
14366 GRAPHICS ON
14367 VIEWPORT 15,115,30,90
14369 !
14370 WINDOW 0,Xmax,-Y1max,Y1max
14371 AXES Xspc/5,2,0,0,5,5,1
14372 AXES Xspc/5,(.1-Y2min)*Y1max/(-Y2min)-Y1max,Xmax,0,5,5,1
14373 CLIP OFF
14374 CSIZE 3
14375 FOR Xlab=0 TO Xmax STEP Xspc
14376 MOVE Xlab-.01,-Y1max-Y1max/10
14377 LABEL USING "#,K";Xlab
14378 NEXT Xlab
14379 MOVE Xmax-Xmax/10,-Y1max-Y1max/5
14380 LABEL USING "#,K";"L."
14381 IF Y1max<10 THEN Y1inc=1
14382 IF Y1max>=10 THEN Y1inc=5
14384 FOR Ylab=0 TO Y1max STEP Y1inc
14385 MOVE 0-.1*Xmax,Ylab-.5
14386 LABEL USING "#,K";Ylab
14387 NEXT Ylab
14388 MOVE -Xmax/7,Y1max+Y1max/10
14389 LABEL USING "#,K";"Kmax-Kmin."
14390 !
14391 IF Y2min<.7 THEN
14392 Y2inc=.1
14393 ELSE
14394 Y2inc=.5
14395 END IF
14396 FOR Rlab=0 TO -Y2min STEP Y2inc
14397 MOVE Xmax+Xmax/15,(Rlab+(Y2min/20)-Y2min)*Y1max*2/(-Y2min*2)-Y1max
14398 LABEL USING "#,K";Rlab
14399 NEXT Rlab
14400 FOR Rlab=0 TO Y2min STEP -Y2inc
14401 MOVE Xmax+Xmax/15,(Rlab+(Y2min/20)-Y2min)*Y1max*2/(-Y2min*2)-Y1max
14402 LABEL USING "#,K";Rlab
14403 NEXT Rlab
14404 MOVE Xmax+Xmax/10,Y1max+Y1max/10
14405 LABEL USING "#,K";"R."
14406 MOVE Gauss(0,1),Gauss(0,4)
14407 FOR Ff=0 TO Ult
```

```
14408 PLOT Gauss(Ff,1),(Gauss(Ff,5 ...
14409 NEXT Ff
14410 MOVE Gauss(0,1),(Gauss(0,5)-Y2min)*2*Y1max/(-2*Y2min)-Y1max
14411 LINE TYPE 4
14412 FOR Gg=0 TO Ult
14413 PLOT Gauss(Gg,1),(Gauss(Gg,5)-Y2min)*2*Y1max/(-2*Y2min)-Y1max
14414 NEXT Gg
14415 !
14416 Q$=""
14417 INPUT "COPY?",Q$
14418 IF Q$="Y" THEN
14419 OUTPUT 2;"N";
14420 GOTO 14425
14421 END IF
14422 IF Q$= N" THEN GOTO 14425
14423 GOTO 14416
14425 SUBEND
```

REFERENCES

No.	Author	Title, etc
1	Shivakumar, V. Forman, R.G.	Green's function for a crack emanating from a circular hole in an infinite sheet. *International Journal of Fracture, Vol.16, No.4,* pp 305-316 (1980)
2	Abramowitz, M. Stegin, I.A.	*Handbook of mathematical functions.* Dover Publications, New York (1965)
3	Newman, J.C., Jr Raju, I.S.	*Stress intensity factor equations for cracks in three-dimensional finite bodies.* Fracture mechanics: Fourteenth symposium – Volume 1: Theory and Analysis, ASTM STP 791 (1983)
4	Edwards, P.R.	*A computer program for the interpolation and extrapolation of crack propagation data.* RAE Technical Report 76115 (1976)
5	Cook, R. Rooke, D.P. Smith, A. Bowles, R.I.	*Residual stress fields at notches: Effect on fatigue crack growth.* RAE Technical Report 85049 (1985)
6	Newman, J.C., Jr Edwards, P.R.	*Short crack growth behaviour in an aluminium alloy – an AGARD cooperative test programme.* AGARD Report AGARD-R-732 (1988)



a) Residual stress distribution without compressive yielding

b) Residual stress distribution after compressive yielding

Fig 1 Typical residual stress distributions showing partitions for interpolation routine

Fig 2

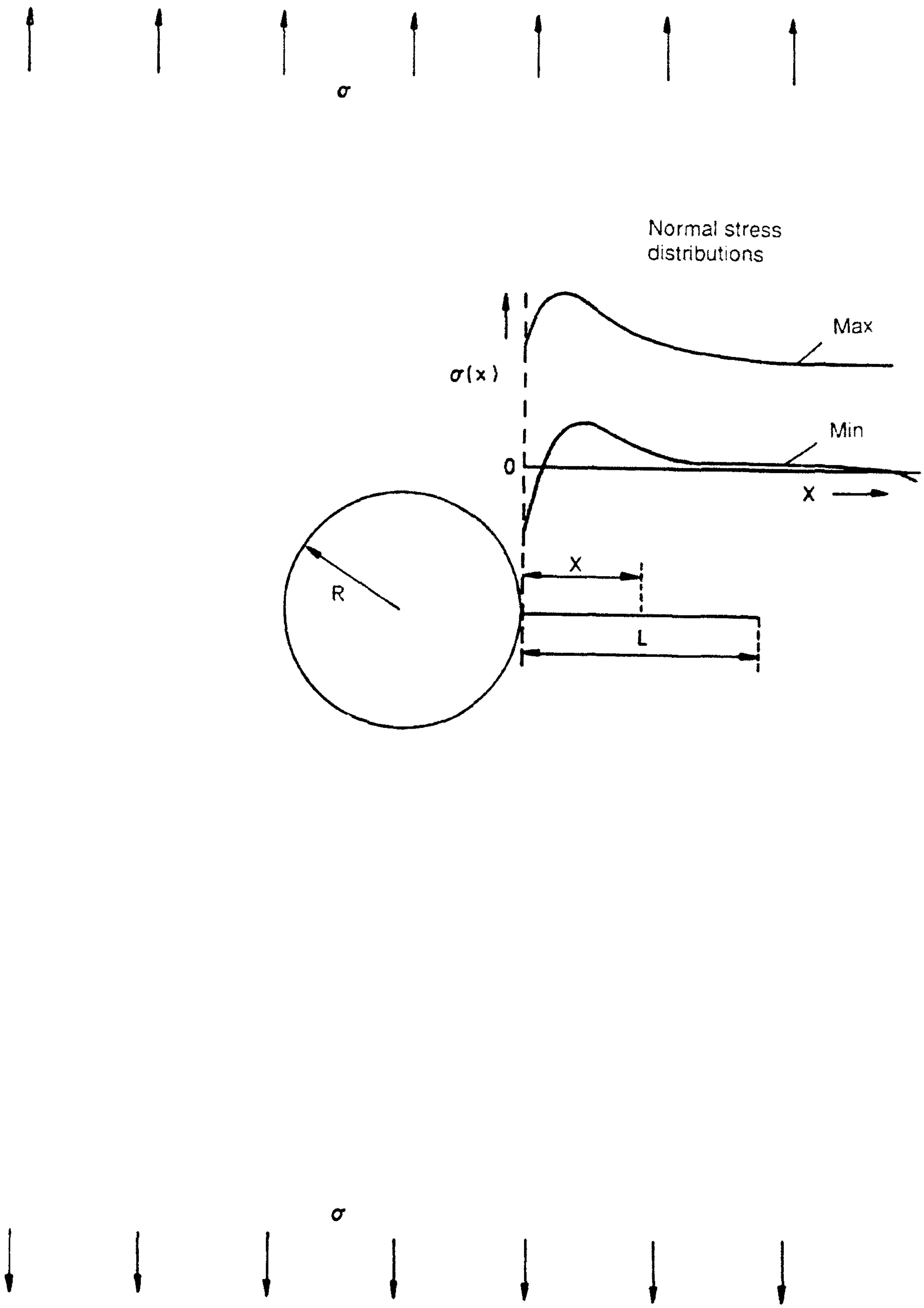


Fig 2 Schematic representation of remotely located cracked hole and resultant stress distribution

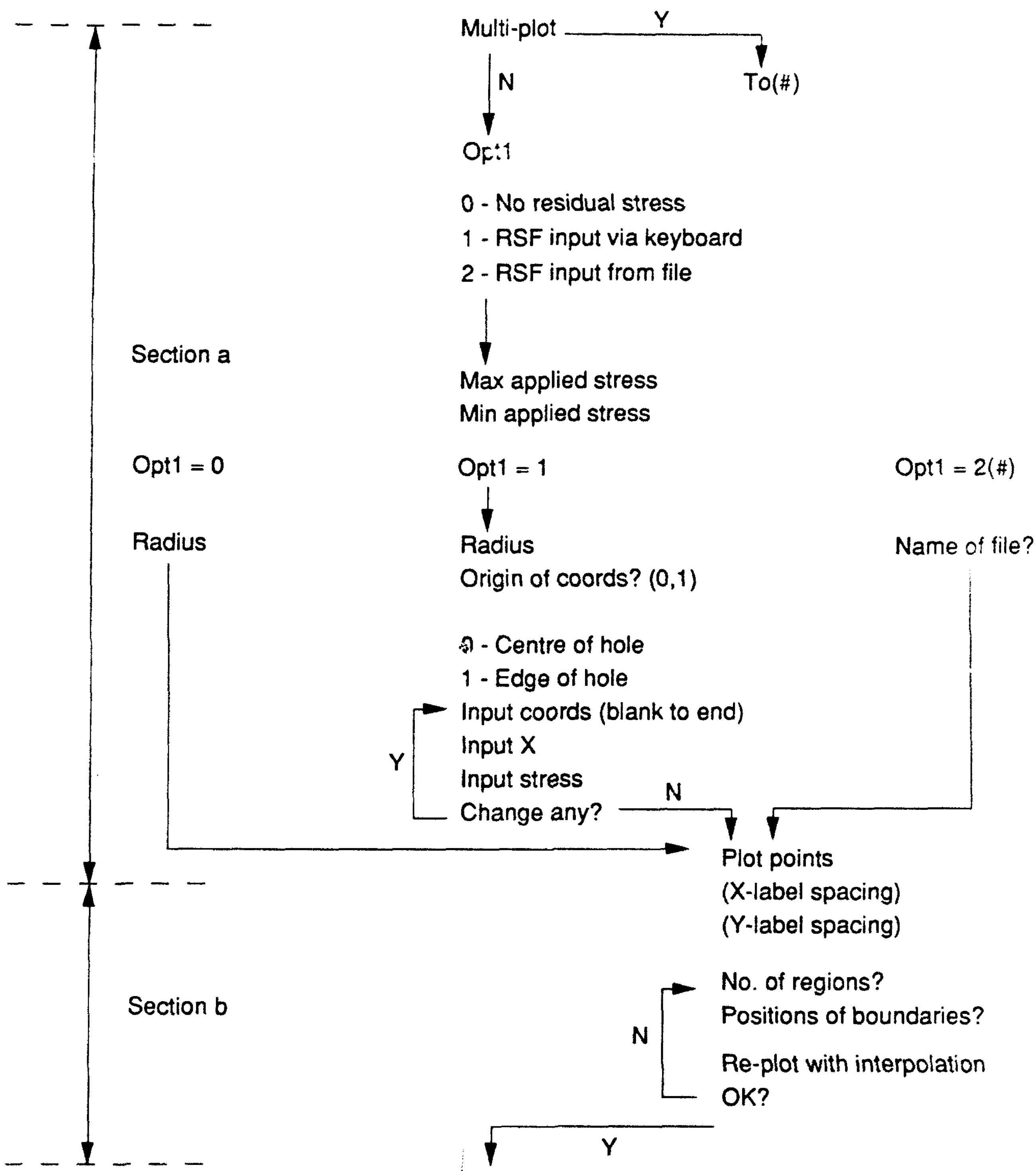


Fig 3 Flow diagram of computer program

Fig 3 (cont)

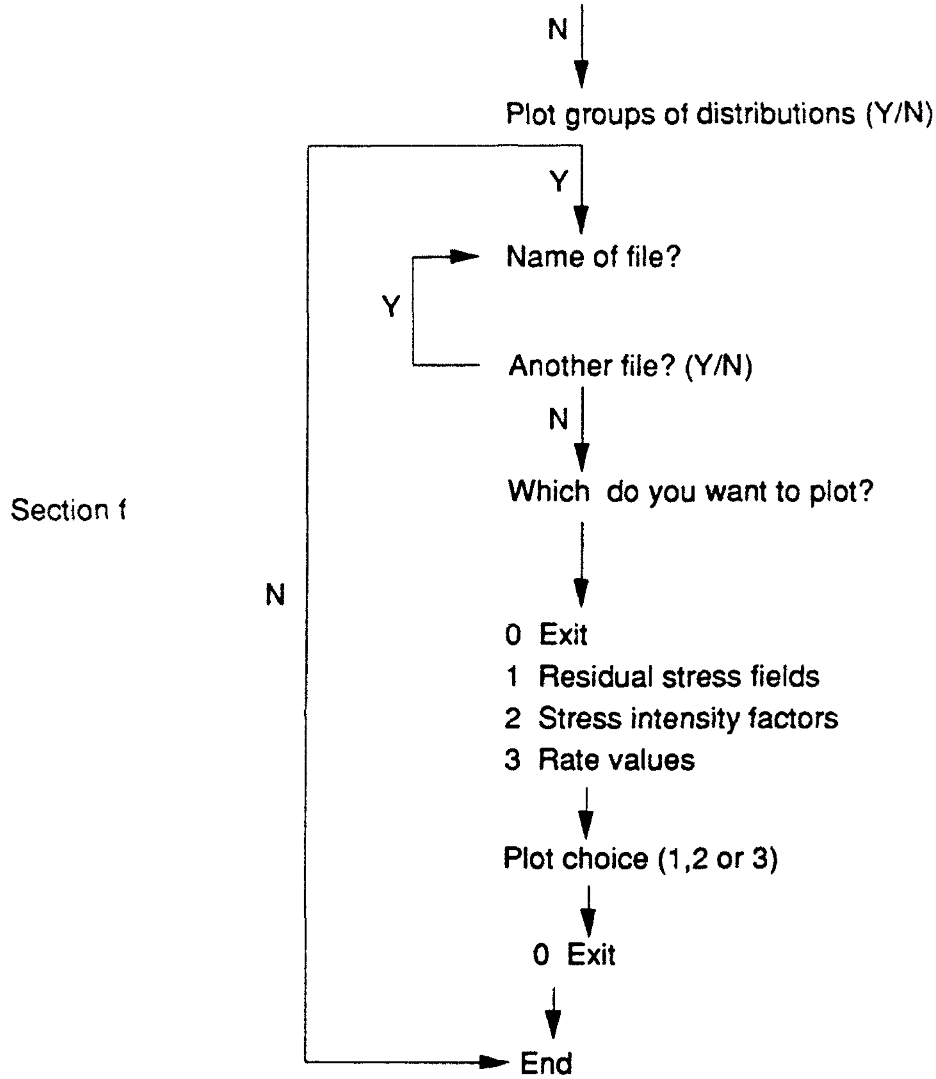


Fig 3 (cont) Flow diagram of computer program

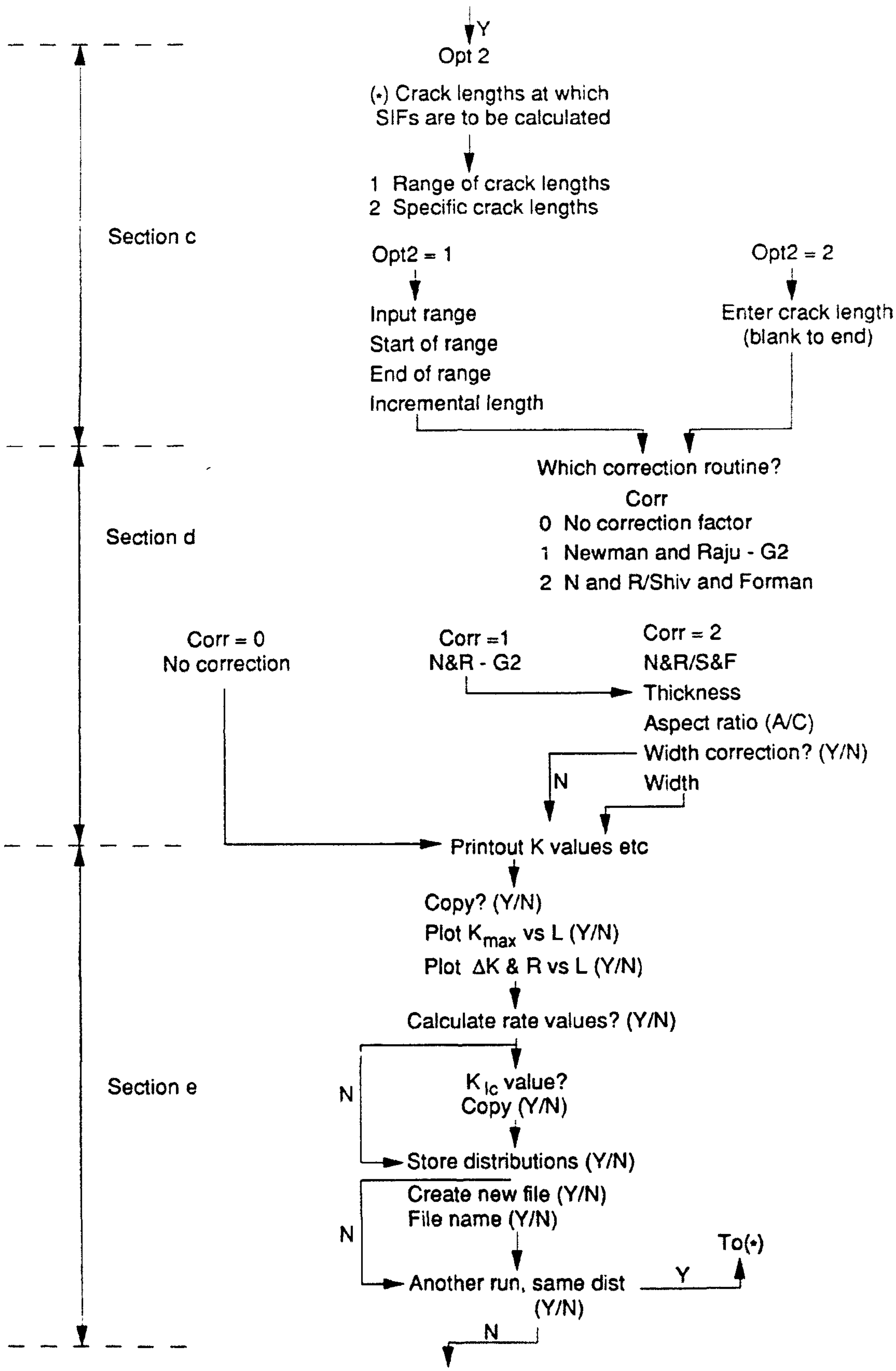


Fig 3 (concluded)

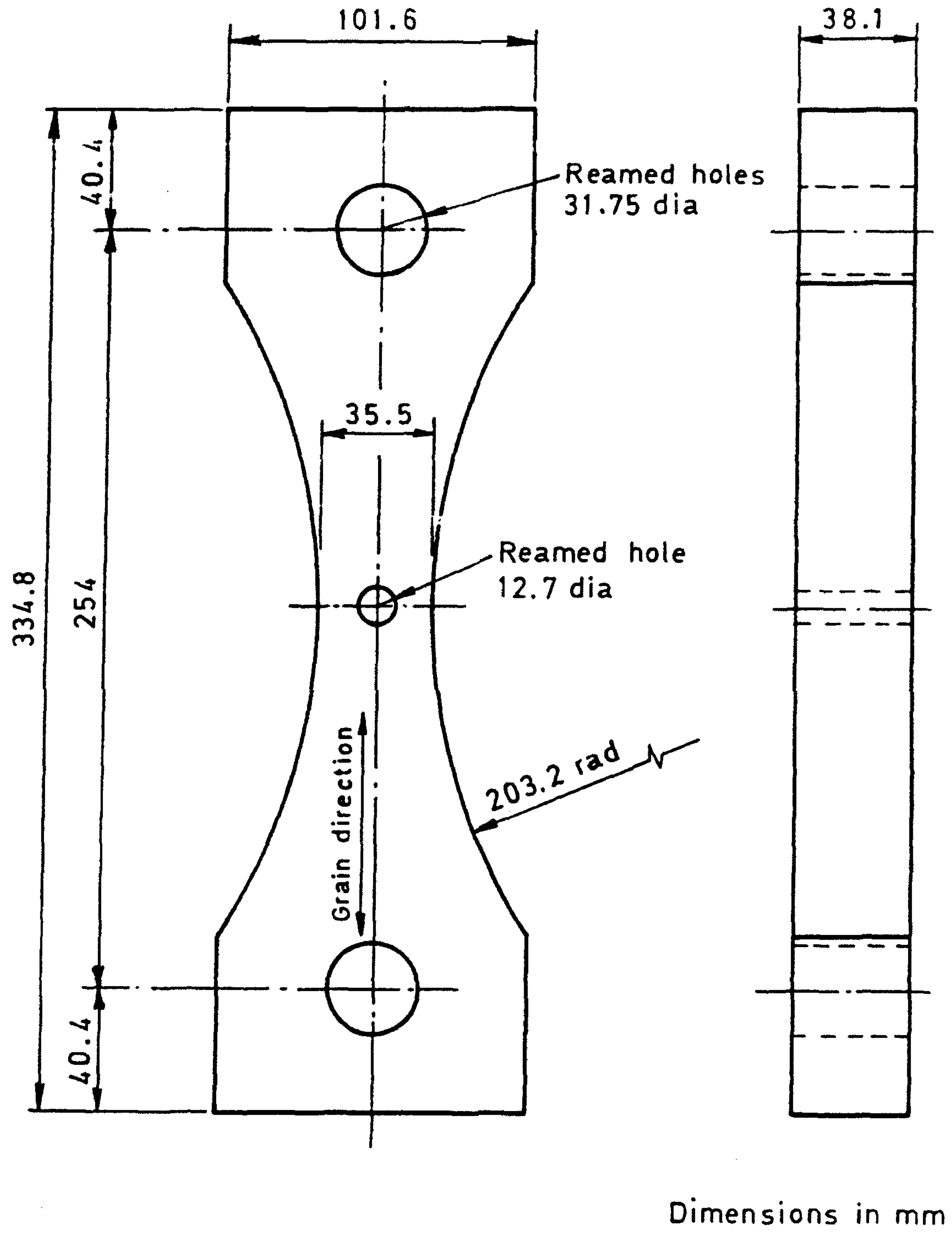


Fig 4 Fatigue test specimen

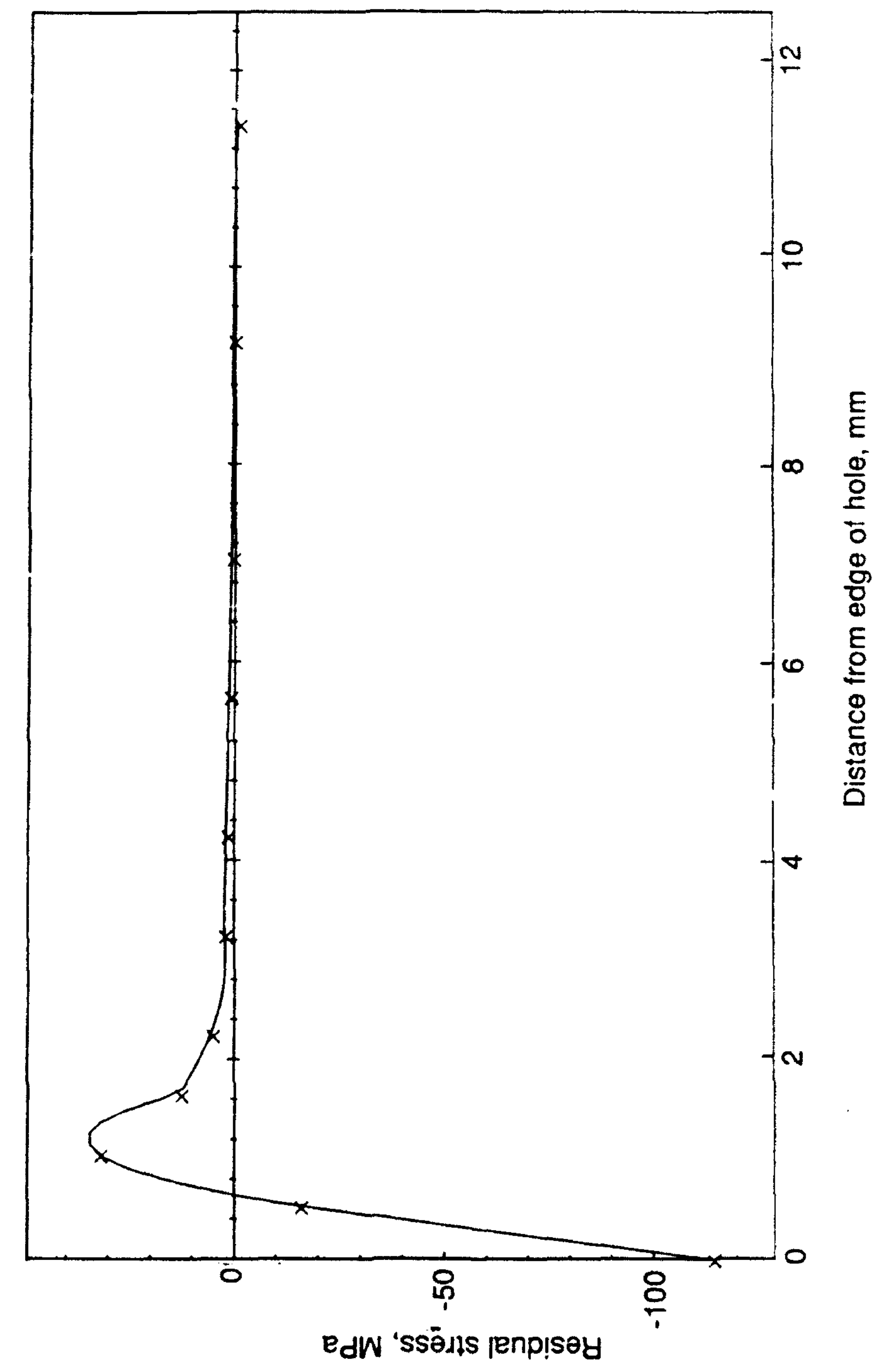


Fig 5 Residual stress distribution for prestress a

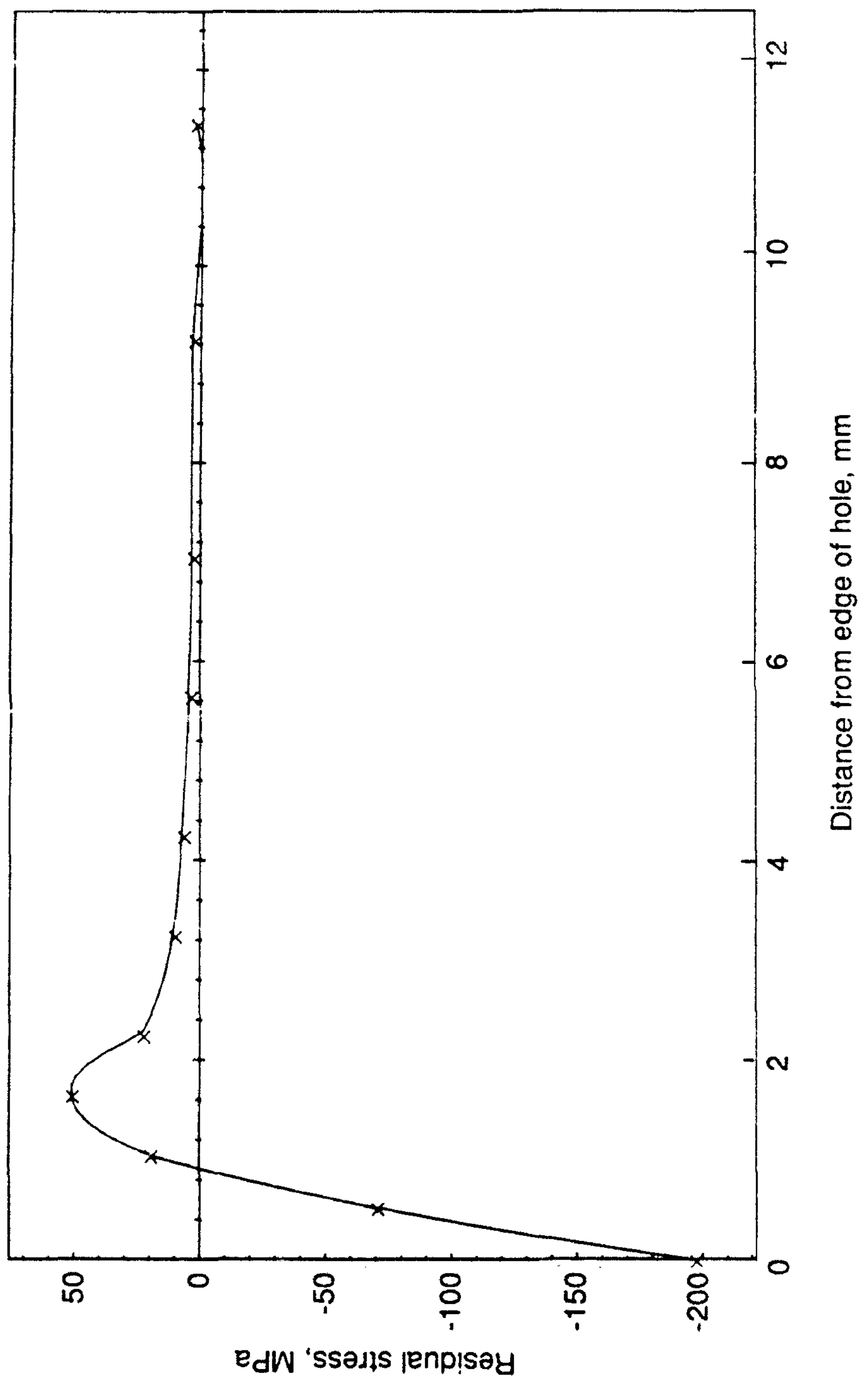


Fig 6 Residual stress distribution for prestress b

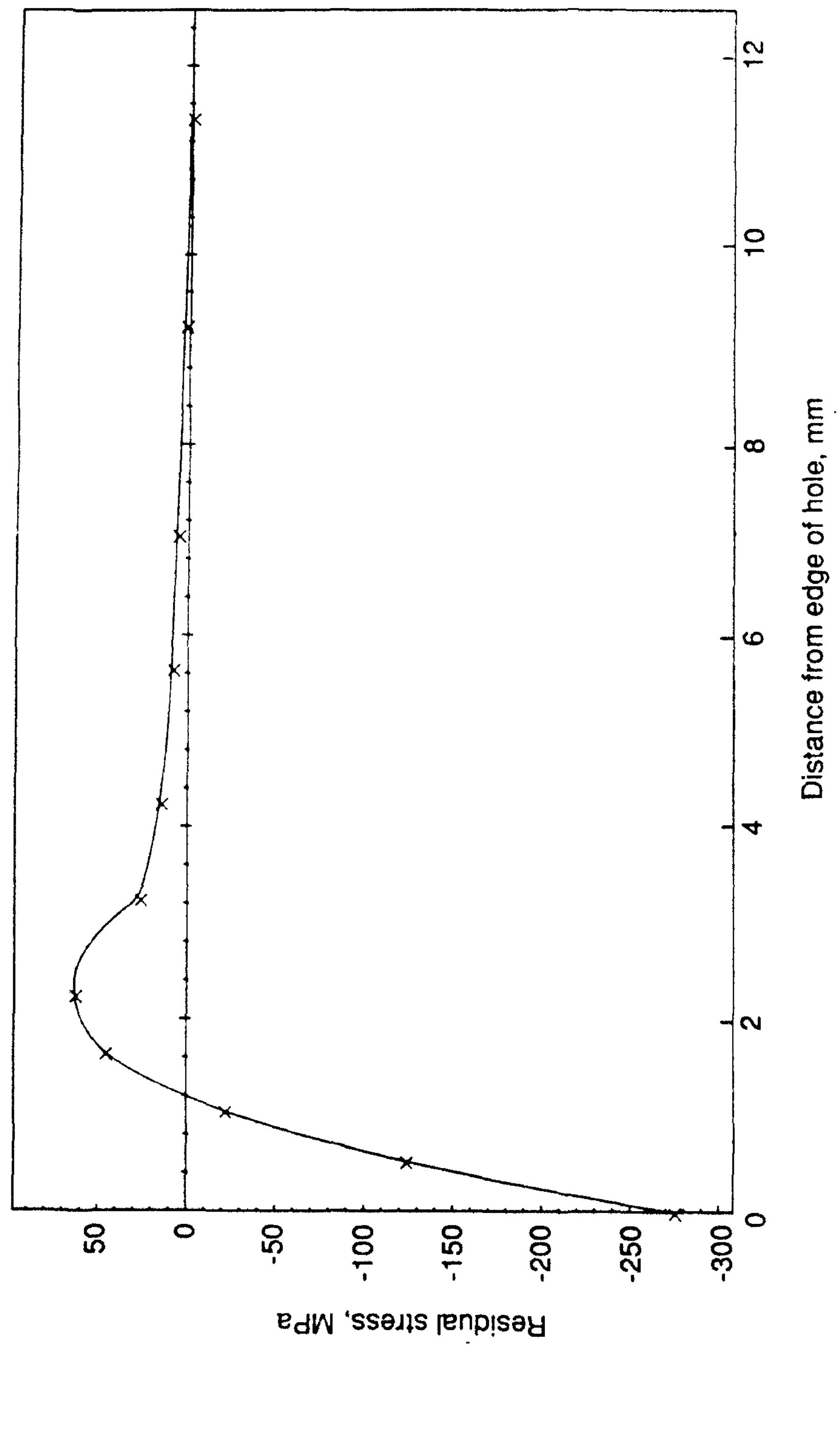


Fig 7 Residual stress distribution for prestress c

Fig 8

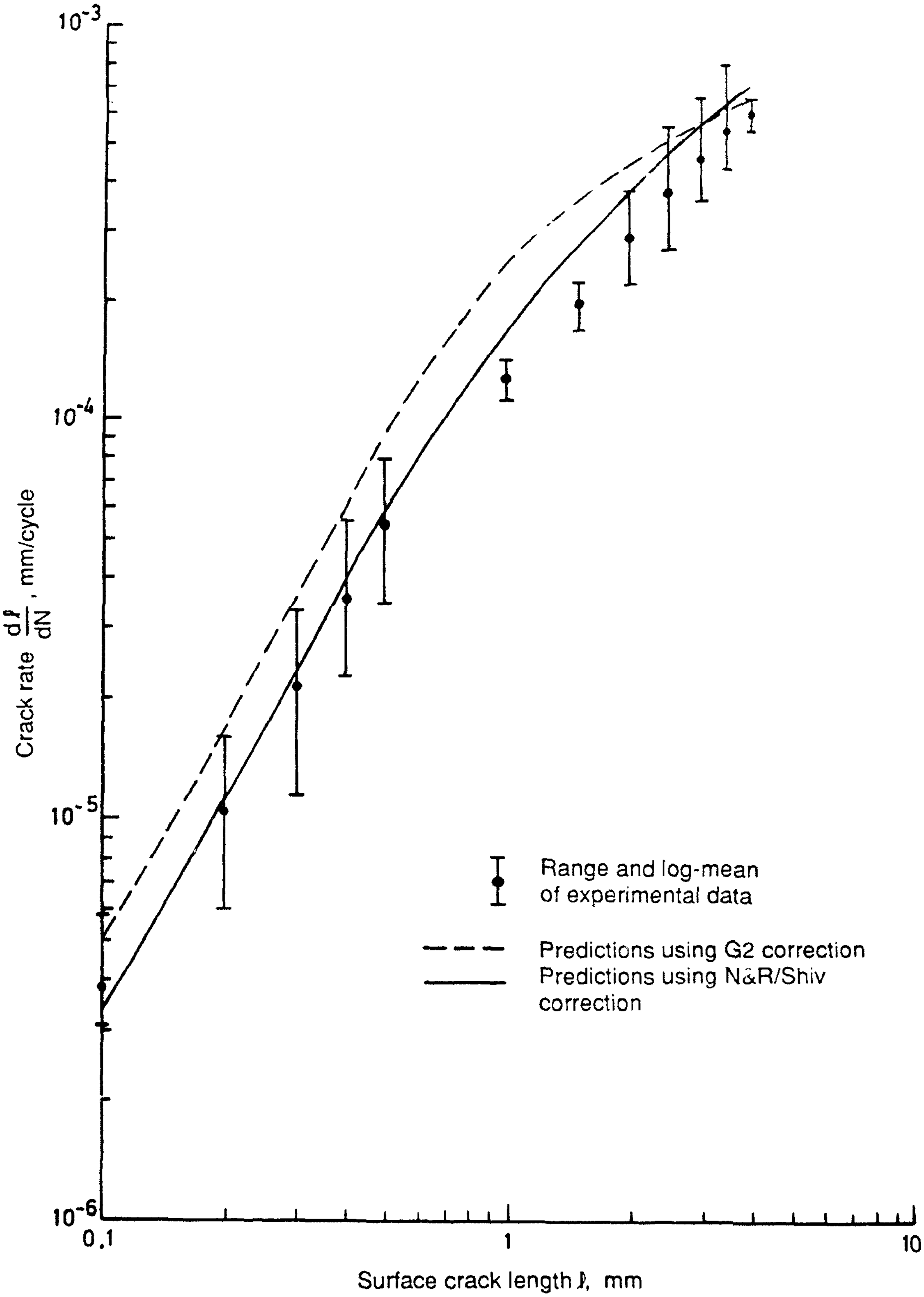


Fig 8 Comparison of predicted and experimental crack propagation rates (prestress a)

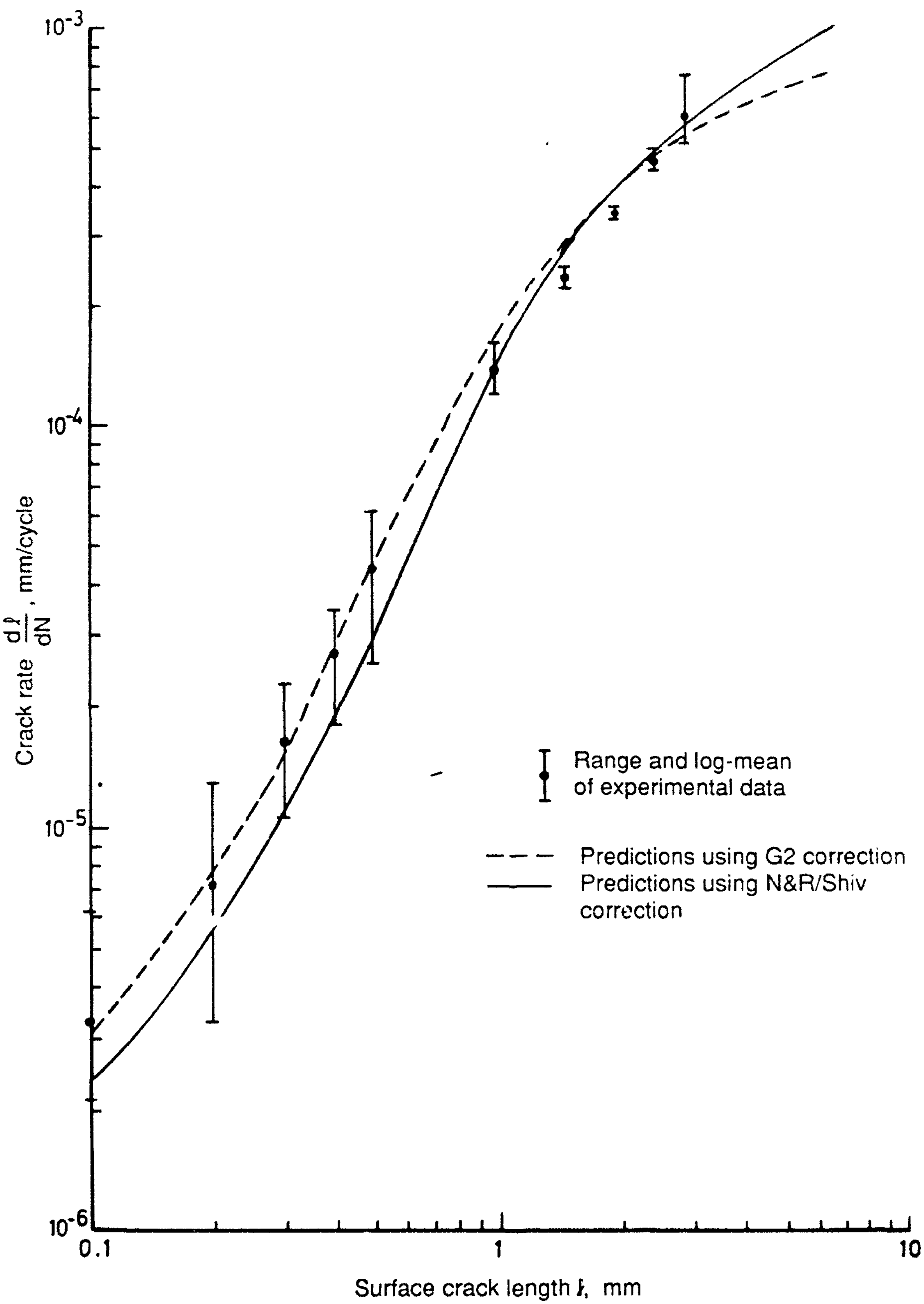


Fig 9 Comparison of predicted and experimental crack propagation rates (p· stress b)

Fig 10

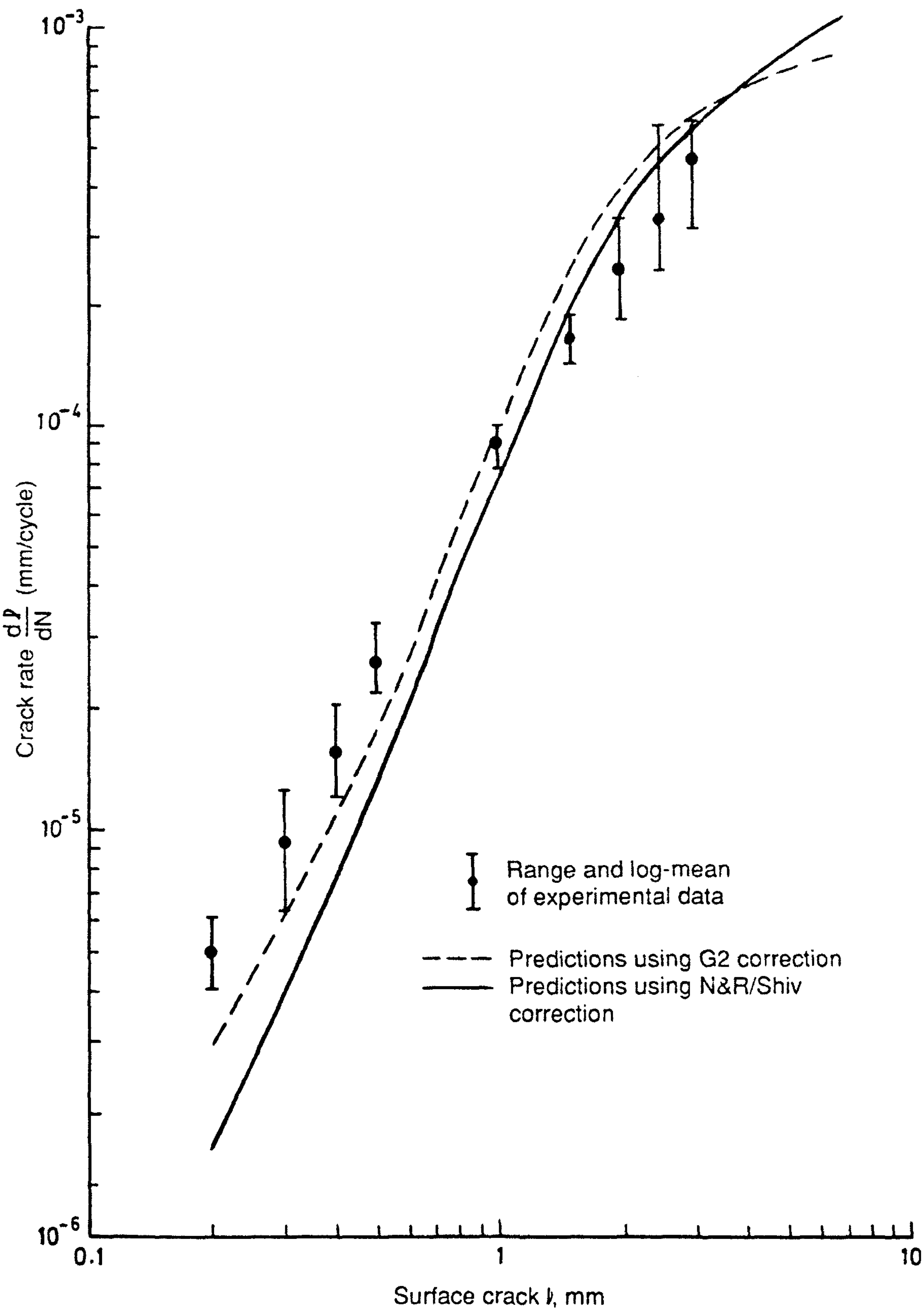


Fig 10 Comparison of predicted and experimental crack propagation rates (prestress c)

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNLIMITED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, *eg* Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference DRA TM Mat/Str 1207	3. Agency Reference	4. Report Security Classification / Marking UNLIMITED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	DRA Farnborough, Hampshire, GU14 6TD
5a. Sponsoring Agency's Code	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title

A computer program to calculate growth rates for cracks at notches in regions of residual stress

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1, Surname, Initials	9a. Author 2	9b. Authors 3,4 ...	10. Date	Pages	Refs
Cook, R.	Dorey, S.	Rooke, D.P.	November 1992	54	6
11. Contract Number	12. Period	13. Project	14. Other Reference Nos.		

15. Distribution statement

(a) Controlled by – Unlimited Distribution

(b) Special limitations (if any) –

If it is intended that a copy of this document shall be released overseas refer to DRA Leaflet No. 3 to Supplement 6 of MOD Manual 4

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)

Fatigue. Crack. Prediction. Program. Fracture mechanics. Residual stress. Cold expansion.

17. Abstract

This Memorandum describes a computer program which can be used to predict the growth rates of fatigue cracks emanating from notches and growing through residual stress fields. The residual stress distributions, alternating loading conditions and specimen geometry must be specified by the user. The program uses a Green's function technique to calculate the stress intensity factor due to the applied and residual stresses for any crack length. The calculated stress intensity factor can be corrected to account for the exact crack shape, if it is known. The crack growth rate is obtained from a database of experimentally determined crack growth data as a function of stress intensity factor, for a number of different materials.